DOCUMENT RESUME

ED 039 778 24 EM 008 133

AUTHOR	Oldehoeft, Arthur E.
TITLE	Computer-Assisted Instruction in Teaching Numerical Methods. Final Report. (A Computer System to Teach Computational Mathematics.)
INSTITUTION	Purdue Univ., Lafayette, Ind. Dept. of Computer Science.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-8-E-010
PUB DATE	Mar 70
GRANT	OEG-0-8-080010-3532
NOTE	282p.
EDRS PRICE	EDRS Price MF-$1.25 HC-$14.20
DESCRIPTORS	*Algorithms, College Students, *Computer Assisted Instruction

ABSTRACT

The design and development of a program of computer-assisted instruction (CAI) which assists the student in learning elementary algorithms of an undergraduate numerical methods course is presented, along with special programing features such as partial precision arithmetic, computer-generated problems, and approximate matching of mathematical expressions. The program, designed to operate under the Purdue Instructional and Computational Learning System (PICLS) language, is described in detail: first, there is a tutorial presentation of the mathematical development surrounding an algorithm, the student then formulates the solutions to several problems to display a working knowledge of the algorithm, and, finally, the student progresses to an exploratory stage where he may formulate the solution to his own problems (all computation is assumed by the computer). An experiment to test this program is also described; results include analyses of student attitudes and performances, cost factors, and efficiency. (Author/SP)

ED039778

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

COMPUTER-ASSISTED INSTRUCTION
IN
TEACHING NUMERICAL METHODS

PURDUE UNIVERSITY

COMPUTER SCIENCES DEPARTMENT

DIVISION OF MATHEMATICAL SCIENCES

EM008133

ED0 39778

FINAL REPORT
Project No. 8-E-010
Grant No. OEG-0-8-080010-3532

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

COMPUTER-ASSISTED INSTRUCTION IN TEACHING NUMERICAL METHODS

March 1970

U. S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

Final Report
Project No. 8-E-010
Grant No. OEG-0-8-080010-3532

COMPUTER-ASSISTED INSTRUCTION IN TEACHING NUMERICAL METHODS
(A COMPUTER SYSTEM TO TEACH COMPUTATIONAL MATHEMATICS)

Arthur E. Oldehoeft
Purdue University
Lafayette, Indiana
March 1970

The research reported herein was performed pursuant to a grant with the Office of Education, U.S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.

U. S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

ACKNOWLEDGEMENTS

I would like to thank Professor S. D. Conte for his guidance and encouragement in this research, Professor R. R. Korfhage for his interest and supervision in developing PICLS, and Professor F. J. Frederick for his helpful suggestions in the preparation of this dissertation.

I am also indebted to Mr. John Steele for providing access to computing facilities in order to obtain experimental results. Special thanks are due Mr. Kenneth Adams who was instrumental in installing permanent file residence features for PICLS and in revising PICLS to maintain operational compatibility with progressive versions of the MACE operating system.

My wife, Margie, also provided invaluable help and encouragement throughout this entire investigation.

TABLE OF CONTENTS

Page

LIST OF TABLES v

LIST OF FIGURES vii

ABSTRACT viii

INTRODUCTION 1

- Current State of the Art 1
- Description of the Research and Development 7
- An Overview of the Hardware and Software 11

AN ANALYSIS OF CONSTRUCTED MATHEMATICAL RESPONSES 13

- Description of the Problem 13
- Preliminary Design Considerations 15
- Description of the Method and Its Limitations 16
- The Subroutine MATCH 24
- Mathematical Justification of the Method 30
- Concluding Remarks 32

DESIGN AND DEVELOPMENT OF THE CAI COURSE 35

- General Philosophy and Design Considerations 35
- Structure of the Tutorial Mode 40
- Structure of the Problem Mode 46
- Structure of the Investigation Mode 51
- Special Program Features 52
- Concluding Remarks 57

EXPERIMENTAL RESULTS AND GENERAL CONCLUSIONS 63

- The Purpose and General History of the Experiment 63
- Characteristics of the CAI and Conventional Groups 66
- An Analysis of Student Performance 69
- Observations of the Proctor 77
- Results of the Questionnaire 79
- The Economics of CAI 82

GENERAL FINDINGS AND RECOMMENDATIONS 89

LIST OF REFERENCES 95

Page

APPENDICES

Appendix A: Student Manual for a Computer-Assisted Course in Computational Mathematics 98
Appendix B: Lesson Plans 160
Appendix C: Questionnaire and Examinations 217
Appendix D: Sample Student Output 234

VITA 272

LIST OF TABLES

Table | Page
1. Available Operators and Functions 16
2. Computational Probability of Failure for m Random Points 19
3. Points Outlining Failure Region 20
4. Predicted and Experimental Probabilities of Failure . . . 20
5. Experimental Probability of Failure for Ten Identities . 21
6. Φ-Tables for Examples 5-8 24
7. Φ-Tables for Examples 9-13 30
8. Mathematics Background for Various Groups 67
9. Rankings by Math Semester Hours (mh) 68
10. Rankings by Math Gradepoint (gp) 68
11. Examination Scores for Various Groups 70
12. Predicted and Actual Performance of the CAI Students . . 71
13. Examination Items with Large Group Differences 72
14. Comparison of CAI-1 with Approximate Representatives in C 75
15. Comparison of CAI-2 with Approximate Representatives in C 75
16. Performance at the Graduate Level 76
17. Distribution of Responses on the Questionnaire 81
18. Average Student and Computer Time Requirements 85
19. Computing Costs for CAI 86
20. Individual Responses to Questionnaire Items 225
21. Rankings by Exam 1 Scores 232

Table Page

22. Rankings by Exam 2 Scores 232

23. Rankings by Exam 3 Scores 232

24. Rankings by Average Scores over Three Exams 233

LIST OF FIGURES

Figure Page

1. Instructional Strategy for Multiple Choice Items . . . 43

2. Instructional Strategy for Constructed Mathematical Responses 44

3. Problem Mode Strategy for Constructed Responses 48

ABSTRACT

Oldehoeft, Arthur Earl. Ph.D., Purdue University, June, 1970. A Computer System to Teach Computational Mathematics. Major Professor: S. D. Conte.

The program is designed to operate under PICLS on a CDC 6500 computer and assists the student in learning elementary algorithms of an undergraduate numerical methods course. The program begins with a tutorial presentation of the mathematical development surrounding an algorithm and a description of the mechanics of the algorithm. The student participates throughout this phase and is required to work numerous exercises. The program then requires the student to formulate the solution to several problems in order to display a working knowledge of the algorithm. Finally, the student progresses to an exploratory stage where he may formulate the solution to his own problems. All computation is assumed by the computer and the student is free from conventional programming and debugging.

The design and construction of this program is presented along with special programming features such as partial precision arithmetic, computer-generated problems, and approximate matching of mathematical expressions.

The first experiment is described in detail. Student attitudes and performances, cost factors, and efficiency are analyzed.

CHAPTER I

INTRODUCTION

Current State of the Art

The possibilities of using a computer to aid in the instructional process have attracted researchers from a variety of backgrounds. The magnitude of interest in computer-assisted instruction (CAI) is demonstrated by the large volume of literature available for public distribution. For recent detailed reviews of the work in CAI, the reader is directed to two recent articles by Feldhusen and Szabo [13,14] and publications by the Entelek Corporation [10,20].

The basic problems which are encountered in CAI are generally attributed to the inability to totally define and control the human learning process and the limited ability to communicate with a computer in a natural language so as to make the computer behave as a human tutor. Kindred [22] classifies the research areas as pedogogical-psychological and technical-practical according to the types of problems encountered.

In the first category, the concern is with the theory of learning and attempts are being made to define and control those variables which would play an active role in a teaching model. The ultimate objective would be to construct a teaching model to adapt to individual differences and lead a given student to a maximum level of performance in the least possible amount of time. Stolurow and Davis [35] describe a somewhat more practical model. Given the student variables along with a minimum acceptable level of performance and a maximum allowable instruction time, the machine would select from a set of teaching strategies that strategy which is most likely to satisfy the constraints of the problem. It is assumed that the machine has a large number of teaching programs at its disposal. Teaching strategies would dynamically change on the basis of the student's performance and personal traits. The success in constructing such a model depends heavily on the ability to define and evaluate the effects of factors such as student variables and their relevance to learning, branching strategies, methods of feedback, and modes of instruction.

Several points are often cited in favor of CAI: self-pacing, cheat proof, immediate feedback and reinforcement, simultaneous teaching, testing, and remedials functions, access to a history of student performances, and the freeing of the instructor's time for counseling. Many of these advantages have not been convincingly demonstrated.

If self-pacing means that a student progresses through a fixed instructional sequence at a rate which is determined by his own abilities and understanding, then the student cannot be delayed by faulty hardware or software, inefficient keyboards, and slow typing mechanisms. Experiments at Penn State [29] point out that CAI is not self-paced in this respect. If self-paced also implies that the student is not exposed to materials already learned, then the student cannot be confined to a fixed instructional program.

To date, very little is known about how one would effectively combine the teaching, testing, and remedial functions. Early attempts are reported by Suppes [36] on the construction of multitrack programs for elementary arithmetic. The nature of the subject matter is drill and practice. The level of difficulty and the amount of drill is determined by a percentage of right and wrong answers.

Although computers can provide a complete history of a student's performance along with prescribed summaries and statistics, it is not clear how researchers, instructors, and administrators can make good use of the information. Also, it remains to be determined how traditionally trained teachers will effectively use their time for other functions such as counseling students. These difficulties are noted by Fein [11].

The cited advantages of CAI are embodied in the concept of individualization. Oettinger and Marks [30] point out that a definition of individualization is not generally agreed upon. Related to the learner and taken literally, a computer system would have to tailor itself to all characteristics of an individual which affect the learning process. However, researchers with experience in this field generally agree that CAI has the potential for a high degree of realization of the cited advantages.

In the technical-practical area, emphasis is placed on the development and evaluation of hardware, CAI programming systems, and actual CAI course material, and on attempts to specify a systematic set of rules for designing instructional material. The theory which supports the existence of this area is that some advantages of CAI over traditional instruction can be demonstrated through a sensible approach.

In the area of CAI software, a number of languages have evolved. Languages which are used for instructional applications have been reviewed by Frye [16] and Entelek [20]. Some of the languages which have been designed primarily for creating course material are PLANIT [12], MENTOR [15], PILOT [38], ELIZA [37], Coursewriter [21] (various versions), PICLS [24], TUTOR [1], and ISL-1 [33].

CAI languages are designed with the intention of providing the course author with a nontechnical method for creating and implementing instructional materials. Embedded in these languages are techniques for processing student responses. Keyword matching and character editing are standard routines and the PLANIT language also has a phonetic analyzer. Although these features are useful and represent an approach to the problem of free communication, their use is left to the ingenuity of the course author. We are still a long way from automatically processing complex natural language responses.

The design and implementation of course material remains a monumental task. A survey by Balough [2] cites a wide range of estimates, varying from eleven to two hundred instructor hours, needed to prepare one hour of student instruction. Charpe and Wye [7] report that more than two hundred total man hours are needed to provide for one hour of student instruction.

Investigators such as Bunderson [4], Childs [8], and Mager [26] have studied the problem of systematically designing course material. They generally agree that certain basic steps are necessary to produce effective results.

1. Specify the terminal objectives of each lesson in terms of
    a. the kind of behavior which will be accepted as evidence that the student has learned,
    b. the conditions under which the desired behavior is expected to occur, and

   c. how well the student must perform in order to have his behavior considered acceptable.

2. Perform a task analysis by
   a. selecting the sequence of learning experiences that are likely to attain the chosen objectives,
   b. specifying all possible outcomes, and
   c. selecting the learning experiences to remedy erroneous outcomes.
3. Program the course materials.
4. Test and revise the materials on the basis of actual performance.

Of the above steps, the task analysis is considered to be the most ill-defined. A selection of learning experiences is based largely on the judgment of the course author. If constructed responses are required, it is difficult and perhaps impossible to specify all possible outcomes even on a single item and provide the appropriate remedial instruction. If two students arrive at the same erroneous answer, they may have done so for reasons unrelated to each other. Without a study of the histories of many students, it is difficult to specify even the probable errors.

In the area of actual development and implementation of college level course materials, very few complete CAI courses actually exist. Several of the major contributors have been the University of Illinois [25], Florida State University [19], the University of Texas [5], and Pennsylvania State University [29]. Based on reports from these institutions, some agreements and discords can be noted.

1. A systematic approach to the development of instructional material is necessary.

2. Achievement and retention comparisons between CAI and traditional instruction have not yielded conclusive results and, on a course by course analysis, there is some disagreement.
3. Comparisons in instruction time have yielded contradictory results. Times are dependent on the nature of the course and the terminal hardware.
4. The majority of CAI students express a favorable attitude toward this method of presentation.

There is also a general agreement that CAI cannot be justified on the basis of cost at this point in time [2,23]. Some inconsistencies or lack of conclusions might be attributed to environmental variations, variations in the types of CAI experiments, and poor measuring devices. Experiments have been performed with various types of hardware, author languages, and teaching strategies. Variations are reported in the method of selecting samples, the size of the samples, and the duration of the experiment. Although experiments have involved a variety of materials, it is generally agreed that the areas which are most natural for CAI are drill and practice, simulation, and problem-solving. In these areas, the computational or repetitive power of the computer is more easily applied.

In view of the general difficulties which exist, many researchers do not consider CAI as the panacea, but rather a component which might play an effective role within a system of educational technology. A broader view appears to call for a total reorganization of the structure of educational institutions, a structure in which the computer is one resource to be used where it is most efficient in the instructional process. The social, political, and economic difficulties involved with moving toward an educational technology are cited by Oettinger and Marks [30]. They conclude that the goals and techniques of education are not yet well enough defined for the realization of a technology. Wilson [39] is concerned with similar problems, but does foresee possible use of a computer in instruction, especially in the areas of mathematics and languages.

Description of the Research and Development

This investigation is concerned with the feasibility of using a computer to aid in the teaching of an undergraduate numerical methods course. In an introductory numerical methods course, or what will be referred to hereafter as computational mathematics, the typical student is a college junior who has just completed the basic sequence in calculus, differential equations, and an introductory course in matrix algebra. He is expected to know a programming language well enough to program computational procedures for a computer. Since the algorithms to be taught are designed and analyzed for computer use, it seems feasible at the outset that the computer itself might aid in the instructional process.

By traditional instruction, the undergraduate student in computational mathematics is faced with several problems.

1. The cumbersome arithmetic associated with a numerical method can discourage the student from working anything more than the simplest type of examples. An intuitive feeling for how an algorithm behaves in practice and a knowledge of its deficiencies often requires working a variety of problems. This

requires both time and effort on the part of the student to either write and debug his own programs or obtain available routines from an established library. Due to limited computer resources and commitments to other courses, it is unlikely that the student can explore more than a handful of methods on the computer.

2. With the traditional mode of classroom instruction, it is difficult to expose the student to the variety of examples and applications needed to demonstrate the deficiencies or power of a method. This is due, in part, to the amount of material which must be presented and also to the cumbersome arithmetic. The instructor usually limits his discussion to the basic theory which establishes the existence of a method, an explanation of the mechanics of the algorithm, and an example or two which can be demonstrated on the blackboard. The examples may not be carried to completion. Textbooks, which present tabulated computer results for particular examples help to remedy the situation, but the student does not normally work through these examples. From this framework of examples, or faith in the instructor, the beginning student is expected to gain an intuitive feeling for a numerical method.
3. Beginning courses usually require a mathematical exposure to elementary calculus. Unless the underlying theorems are elementary in nature, they are at most stated in passing. As a result, a great deal of emphasis is placed on a description of the mechanics of an algorithm. With the traditional mode of instruction, the student does not participate in the development

of the mechanics and may not have the time or computer resources to practice the application of all algorithms.

The problems stated above can be more or less attributed to the nature of a computational mathematics course. In addition, the student faces problems which are common to all courses.

4. Individual attention is given in the classroom to only those students who interject comments or ask questions. Verbal communication is normally attempted by only a small percentage of the class.
5. The lecture is prepared for the level of the average student. The better students are unmotivated and the weaker students are slow in grasping the material.
6. Due to administrative demands or research interests, the instructor cannot devote sufficient time to counseling students.

The work reported in this paper does not prescribe a cure for these difficulties since it is not known how to program an ideal teacher. However, the possibility of reducing the severity of some of the problems can be explored by constructing, implementing, and testing a computer-based instructional system for computational mathematics. The primary objectives of this investigation are concerned with feasibility and are stated as follows:

1. design and implement a CAI program to teach computational mathematics and investigate the technical difficulties associated with constructing and using such a system;
2. implement techniques which might be useful in an attack on problems 1-3 stated above; and

3. experiment with the system in an attempt to determine student acceptance and compare this method with the conventional method of instruction.

To accomplish these objectives, twenty-five lessons were designed for computer presentation. The first teaches the use of the system while the remaining twenty-four cover a variety of numerical methods which the student would study in conjunction with outside reading assignments from Conte [9], the textbook currently used at Purdue. In a direct attack on problems 1-3 stated above, each lesson consists of a tutorial mode, a problem mode, and an investigation mode. In an attempt to free the student for concentration on the development and application of algorithms, the burden of cumbersome arithmetic is assumed by the computer and detailed programming is not required. Problems 4-6 are approached in a manner similar to that in other CAI courses described in the literature. An overall description of the design, implementation, and associated difficulties is presented in Chapter III. Chapter II is devoted to a description of special features needed to handle mathematical expressions entered by the student from the CAI terminal. The results of an initial experiment conducted during the Fall Semester of 1969 are reported in Chapter IV in the form of some numerical measurements and personal observations.

The end product of this development has several uses. First of all, it can serve as a research vehicle for future tests of the effectiveness of CAI. As experience is gained, the system should grow in size and sophistication to incorporate multilevel sequences of instructional material for the purpose of accelerating or decelerating students. Secondly, the system may serve as a self-instructional course for students wishing to study computational mathematics. Finally, in conjunction with traditional lectures, a student may use the tutorial modes for review or remedial work or elect to work graded problems or problems of his own choice.

An Overview of the Hardware and Software

The computational mathematics course was written in the language of PICLS [24], the instructional system available for the CDC 6500. PICLS is designed to operate in an interactive mode under the MACE Operating System [32] for the CDC 6500 at Purdue University.

The MACE Operating System with interactive facilities was developed by Purdue Computing Center personnel. A typical request for service from a CAI terminal is assigned a high priority by MACE causing lower priority jobs to be rolled out of core long enough for PICLS to service the request. PICLS is then automatically rolled out in order to free core for other jobs. Thus, the response time at a CAI terminal is highly dependent on the status of MACE and the current job mix.

In support of the CAI project in computational mathematics, a set of arithmetic routines was installed in PICLS during the summer of 1968. These routines enable a course author to accept student initiated expressions which could be compiled and evaluated or saved for later evaluation. Special routines were also added to test the equivalence of two mathematical expressions. Due to these special arithmetic requirements, the current version of PICLS is the only version under which the course in computational mathematics is guaranteed to be operational. Subsequent versions of PICLS may not contain those special features mentioned above or described elsewhere in this report.

The twenty-five lessons in computational mathematics consists of approximately 27,000 PICLS instructions. Throughout the development, a linear mathematical notation was employed for suitable use on a KSR-33 Teletype terminal. At instruction time, the course material and PICLS resides on approximately 200 tracks of Control Data 808 disk storage in the form of permanent files. Thus, a student at a CAI terminal may initiate any section of any lesson at any time. The actual hardware requirements for the lesson material are the same as the requirements of PICLS. For the most part, PICLS is written in Fortran but some of the file handling routines are written in machine language. In addition, portions of the instructional material depend on the sixty bit binary word of the CDC 6500 and are not directly transferable to other machines without revision.

CHAPTER II

AN ANALYSIS OF CONSTRUCTED MATHEMATICAL RESPONSES

Description of the Problem

An area of major concern in instructional systems is the design of techniques for processing student responses. While it is desirable to provide the student with complete freedom in responding to questions, techniques are not available for grading free form answers. Phonetic encoders and keyword matching routines are attempts to provide more flexibility in processing constructed answers. In one sense, mathematical responses present a very serious problem since an expression can usually be correctly represented in an infinite number of ways. On the other hand, it is the very concept of equivalence over the real or complex numbers which provides for the development of powerful techniques. The problem of deciding when two expressions are equivalent has been encountered in other applications. Both algebraic and numeric approaches to this problem have been reported.

In a direct algebraic approach using normal and canonical forms, Caviness [6] considers the expressions generated by the rationals and the complex number i, the variables $x_1,\ldots,x_n$, the operators +, -, *, unnested composition, and functions exp, sin, cos, tan. An expression in this class can be reduced to normal form P/Q where P and Q are canonical. This yields a technique for deciding equivalence. Caviness also cites some negative results by D. Richardson. Richardson considers the class of expressions generated by the rationals, Π, ln 2, the variable x, the operators +, *, nested composition, and the functions exp, sin, abs. For this class, the predicate "g≡0" is recursively undecidable. Thus, we have an indication of lower and upper bounds on what can be expected from exact techniques.

In a combined algebraic and numeric approach, Martin [28] uses a hash code assignment scheme to map the set of infinite expressions into a finite number field. For addition and multiplication, range problems associated with floating point arithmetic can be avoided by performing the arithmetic in a finite field. However, exponentiation does not preserve equivalence.

The PLANIT system [12] uses a straight numeric approach by assigning prime integers, starting with 3, to each distinct variable and comparing the resulting values of the expressions. By this technique, $f(x)=x$ and $F(x)=6-x$ would be considered equivalent. This is a simple example of the danger encountered in using numbers.

The method installed in a special version of PICLS for the computational mathematics course consists of a combination of random evaluation and operator analysis. Although random evaluation was considered unstable by Martin for his application, there is some promise in CAI since the correct expressions are known when the material is developed. Also, students are likely to construct answers within the context of the discussion. The purpose of this chapter is to describe this method and analyze its deficiencies. Much of the information stated here has been previously reported by this author elsewhere [31].

Preliminary Design Considerations

In designing a method which seemed suitable for conversational use in CAI, several factors were considered. First, a matching algorithm should have low central processing time requirements in order to avoid any significant increase in the already present system overhead time. Secondly, the probability of failure should be remote. If it should happen that the method fails, then it should normally be possible for the student to enter the same answer with a low probability that the method will fail again. This implies that the variables involved in both the correct answer and the student's answer be treated in some independent sense from one application of the matching process to the next. Finally, the method should be sophisticated enough to be "student proof" if at all possible. From an external point of view, it should be difficult for a student to determine the method of testing equivalence in order to avoid deliberate attempts to fool the algorithm.

The rules adopted for constructing mathematical expressions are similar to those used in Fortran. The student, however, is restricted to the use of variables which have meaning within the context of the discussion and have been defined by the author. Brackets [and] are used to delimit subscript expressions and the operators and functions must be chosen from the two classes Φ or T_1 given in Table 1. The choice of notation was based on the student's assumed knowledge of Fortran, the linear notation imposed by the teletype terminal, and the content of the actual course material in computational mathematics.

Table 1. Available Operators and Functions

T_1: +, -, *, /, composition, ↑(or **) to an integer power, sin, cos, tan, csc, sec, cot, exp, r^x where $r > 0$, sinh, cosh, tanh, csch, sech, coth

Φ: arcsin, arccos, arctan, ln, log(base 10), sqrt, ↑(or **) variable base to a variable or fractional power, abs

In order to allow the student maximum flexibility in constructing responses, it is assumed that a course author will define and maintain the status of variables internally as they are introduced to the student on the teletype page at instruction time. For example, if the variable x is introduced during the course of discussion, then the course author also defines the variable x internally and treats it as an indeterminate over the real field by assigning a random value to it. If x assumes a particular value, the course author must assign the same value to x internally and compute all variables depending on x. In this way, the student may construct responses using any variable which is meaningful within the context of the current instructional material. Examples of how the course author provides this flexibility appear in Chapter III.

Description of the Method and Its Limitations

Any expression which is constructed from defined variables, constants, and the T_1-operators listed in Table 1 will be called a T_1-expression. If an expression contains a Φ-operator, it is not a T_1-expression. For example, sin(x+cos(y)) is a T_1-expression from R^2 to R^1. Arcsin(abs(x)/(1+x**2)) is not a T_1-expression.

Throughout this discussion, the correct expression specified by the course author will be denoted by f while F will denote the student's response. Small letters x, y, x_1, x_2, etc. will denote real variables while capital letters X, Y, X_1, X_2, etc. will denote randomly selected values which have been assigned to the variables. R^n will denote n-dimensional real space. Considering f and F as functions from R^n to R^1, points X_i in R^n, $i=1,\ldots,m$, are randomly chosen and the values $f(X_i)$ are compared with $F(X_i)$. If, for all i, the values are equal, the conclusion is $f\equiv F$. Otherwise $f\not\equiv F$. If both f and F are T_1-expressions, the selection of only one $X\in R^n$ is justified in a succeeding section. There is a 0-probability of selecting X where f and F are not defined. If $f\not\equiv F$, there is a 0-probability of selecting X where $f(X)=F(X)$. If Φ-operators are present in either expression, the 0-probability condition may not hold. The effect of Φ-operators will be discussed later.

Since evaluation is performed on a computer, we can only hope to approximate the 0-probability condition. The method will suffer from the common defects of (1) round-off error, (2) loss of significance, and (3) a possible positive probability simply due to a finite set of computer numbers. As a result, it is possible that two nonequivalent expressions will be judged equivalent or equivalent expressions will be judged nonequivalent. The numerical approach is to approximate equivalence by concluding that $f(X)\equiv F(X)$ if any one of three conditions is satisfied for an error tolerance $\delta=5\cdot10^{-11}$.

(1) $|f(X)| < \delta$ and $|F(X)| < \delta$

(2) $|f(X)-F(X)| < \delta$

(3) $|(f(X)-F(X))/f(X)| < \delta$

In an attempt to avoid range problems such as overflow and loss of significance, the programmer should restrict the selection of random points to a finite interval I based on the structure of the correct answer f. For example, $f(x)=\cos(x)+\sinh(x)$ is computationally equal to $\sinh(x)$ for large $|x|$ since the cos is completely dominated by sinh. In this case the programmer would choose I to be a relatively small interval about 0 to retain the effects of the term $\cos(x)$. The choice of I remains somewhat ill-defined since, once I is known, one can deliberately construct expressions F which will emphasize the computational defects.

Rather than a 0-probability, we have an ϵ-probability where ϵ depends on f, F, δ, I, and the precision of the computation. An a priori estimate of ϵ is not available since the student's answer F is not known. On this basis, several strategies are possible. One strategy would be to conclude $f \equiv F$ if the two functions agree at all m points and conclude $f \not\equiv F$ if they fail to agree at any one point. Another strategy would be to conclude $f \equiv F$ if they agree at any one of the m points and conclude $f \not\equiv F$ if they disagree at all m points. Table 2 shows the probabilities of success and failure for these two strategies. If $f \equiv F$, the first strategy is a poor choice for large m since the probability of success $(1-\epsilon)^m$ tends to zero. It is, however, a good strategy when $f \not\equiv F$ since the probability of failure ϵ^m tends to zero. On the other hand, the second strategy is a good choice when $f \equiv F$ and a poor choice when $f \not\equiv F$. One could choose mixed strategies as alternatives.

In actual practice in instructional settings, the case when $f \not\equiv F$ seemed less susceptible to failure than the case where $f \equiv F$. On this basis, it would appear that the second strategy is the better one for this application. In order to further investigate the instability, the

Table 2. Computational Probability of Failure for m Random Points

Possible Computational Events	when $f \equiv F$ $\Pr\{X_i \in I : F(X_i) \neq f(X_i)\} = \epsilon$	when $f \not\equiv F$ $\Pr\{X_i \in I : F(X_i) = f(X_i)\} = \epsilon$
$F(X_i) = f(X_i)$ for all $i = 1 \ldots m$	$(1-\epsilon)^m$	ϵ^m
$F(X_i) \neq f(X_i)$ for any i	$1-(1-\epsilon)^m$	$1-\epsilon^m$
$F(X_i) \neq f(X_i)$ for all $i = 1 \ldots m$	ϵ^m	$(1-\epsilon)^m$
$F(X_i) = f(X_i)$ for any i	$1-\epsilon^m$	$1-(1-\epsilon)^m$

first strategy was adopted in the program. In order to minimize the probability when $f \equiv F$, the value $m=1$ is used. In other words, the decision for equivalence of two T_1-expressions is based on evaluation at exactly one randomly selected point. Examples 1 and 2 presented below illustrate the possible computational difficulties when $f \equiv F$.

Example 1--Loss of Significance. Suppose the correct solution of $x^2+bx+c=0$ is specified by $f(b,c)=.5(-b+\text{sqrt}(b^2-4c))$ and the student's answer is $F(b,c)=-2c/(b+\text{sqrt}(b^2-4c))$. In theory $f \equiv F$, but using single precision on a CDC 6500 with a computational error tolerance, we have $|1-F/f| > 5 \cdot 10^{-11}$ in a region where $|c|$ is small compared to $|b|$. The magnitude of c which outlines this region was approximated for selected values of b. These values appear in Table 3 and yield the relationship $c = \pm 10^{-5} b^2$. More important than the accuracy of the approximations is the fact that, as $|b|$ increases, $|c|$ increases at a faster than linear rate. If (b,c) is selected in the region between the curves $c = \pm 10^{-5} b^2$, then the incorrect decision $f \not\equiv F$ is made. Sampling from a square with center 0 and side length 2S, excluding the region where $b^2 < 4c$, the probability P of an incorrect decision can be found by integration.

Case 1: If $0<S\leq 4$, $P=8(10^{-5})S/(12+S)$

Case 2: If $4<S\leq 10^{5}$, $P=(10^{-5})S^{1.5}/(3S^{.5}-2)$

Case 3: If $S>10^{5}$, $P=(3S^{.5}-2(10^{2.5}))/(3S^{.5}-2)$

Based on the above formulas, Table 4 shows how P increases with S. For the same values of S, an experimental probability P* was computed based on 10,000 random points.

Table 3. Points Outlining Failure Region

$\|b\|$	10^{5}	10^{4}	10^{3}	10^{2}	10	1	10^{-1}	10^{-2}	10^{-3}
$\|c\|$	10^{5}	10^{3}	10	10^{-1}	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-11}

Table 4. Predicted and Experimental Probabilities of Failures

S	1	10	10^{2}	10^{4}	10^{6}
P	$.61\cdot 10^{-5}$	$.42\cdot 10^{-4}$	$.36\cdot 10^{-3}$	$.33\cdot 10^{-1}$	.79
P*	0	0	$.7\cdot 10^{-3}$	$.45\cdot 10^{-1}$	.41

Example 2--Miscellaneous Expressions. The method of comparison at randomly selected points was tried on ten trigonometric identities* used by Martin [28].

(1) sin(x)tan(x)+cos(x)=sec(x)

(2) (sin(x)cot(x)+cos(x))/cot(x)=2sin(x)

(3) $\csc^{2}(x)+\cot^{2}(x)+1=2/\sin^{2}(x)$

(4) cos(x)cot(x)+sin(x)=csc(x)

(5) (1-sin(x))(sec(x)+tan(x))=cos(x)

(6) sin(x)/(1-cos(x))=tan(x)/(sec(x)-1)

(7) $\csc^{4}(x)-\cot^{4}(x)=\csc^{2}(x)+\cot^{2}(x)$

---

* In identity (10), "abs" was added.

(8) $\mathrm{sin}(x)/(\mathrm{sec}(x)+1)+\mathrm{sin}(x)/(\mathrm{sec}(x)-1)=2\mathrm{cot}(x)$

(9) $\mathrm{cos}^6(x)+\mathrm{sin}^6(x)=1-3\mathrm{sin}^2(x)\mathrm{cos}^2(x)$

(10) $\mathrm{sqrt}((\mathrm{sec}(x)-1)/(\mathrm{sec}(x)+1))=\mathrm{abs}((1-\mathrm{cos}(x))/\mathrm{sin}(x))$

For several arbitrary intervals, the results of evaluation at 10,000 random points are reported in Table 5. L denotes the total length of a symmetric interval about 0 from which the points were selected. The error tolerance $\delta=5\cdot 10^{-11}$ was used for equivalence.

Table 5. Experimental Probability of Failure for Ten Identities

	Length of Interval L					
Case	2	6	18	50	70	90
1	0	0	.0001	.0010	.0020	.0010
2	0	0	.0001	.0007	.0002	.0004
3	0	0	.0002	.0009	.0003	.0004
4	0	0	.0002	.0004	.0004	.0008
5	0	.0020	.0055	.0098	.0112	.0143
6	.0098	.0028	.0040	.0036	.0034	.0037
7	.0314	.0115	.0289	.0556	.0752	.0775
8	.0079	.0030	.0053	.0049	.0048	.0048
9	0	0	0	0	0	0
10	0	0	.0017	.0021	.0023	.0018

Assuming no computational difficulties, one still cannot arbitrarily apply this method to any expressions. As previously mentioned, the 0-probability condition may not hold in theory if one uses the Φ-operators from Table 1. The inverse operators introduce branch lines in the complex plane and when restricted to the reals, disjoint regions may be introduced, any or all of which may be of interest. The abs operator also serves to partition the real line into disjoint regions. The presence of Φ-operators in an expression can be detected when evaluation takes place. Example 3 illustrates how Φ-operators may introduce multiple regions. From an analysis of f, one can usually determine the regions of interest in R^n and the normal approach would be to sample in each region. The difficulty arises in trying to mechanically determine the regions introduced by Φ-operators in F. Example 4 shows that a total disregard of F may or may not yield the correct decision.

Example 3. Let $f(x)=\ln(x^2)+\text{abs}(2-x)$. The term $\ln(x^2)$ introduces two regions $L_1=\{x:x>0\}$ and $L_2=\{x:x<0\}$. The term $\text{abs}(2-x)$ introduces regions $L_3=\{x:x<2\}$ and $L_4=\{x:x>2\}$. L_2 is of interest if we wish to distinguish between identities such as $\ln(x^2)$ and $2\ln(x)$ which hold only on the principal branch. The resultant regions are $D_1=\{x:x<0\}$, $D_2=\{x:0<x<2\}$, and $D_3=\{x:x>2\}$.

Example 4. Let $f(x)=\text{abs}(x)$ and $F(x)=x$. An analysis of f yields the two regions $D_1=\{x:x>0\}$ and $D_2=\{x:x<0\}$. Selecting an X in the latter region detects $f\neq F$. Reversing the roles of f and F, let $f(x)=x$ and $F(x)=\text{abs}(x)$. Now, an analysis of f yields one region $D_1=\{-\infty,\infty\}$ since f is a T_1-function. If we randomly select X from any interval symmetric about 0, there is a $\frac{1}{2}$-probability of detecting the fact that $f\neq F$.

We cannot restrict our attention only to the effects of abs since the standard inverse operators may be used to simulate these operators on R^n, e.g. $\exp(\frac{1}{2}\ln(x^2))=\text{abs}(x)$. The approach taken here is to check for the resolvability of two expressions. In particular f and F are said to be resolvable if the occurrence of a Φ-operator (with argument h) in the expression f implies the occurrence of the same Φ-operator (with same argument h) in the expression F and vice versa. The arguments h are checked for equality by the usual method of random evaluation while the Φ-operators are matched symbolically. During the process of evaluation, the operators and the numerical values of the arguments are recorded in a ϕ-table. Examples 5-8 illustrate this method.

Example 5--Resolvable Case where $f\equiv F$. Let f(x)=sin(abs(x-1)+ ln(y^2)) and F(x)=sin(abs(x-1))cos(ln(y*y))+cos(abs(x-1))sin(ln((y-1) (y+1)+1)). An analysis of f yields the following regions in R^2: $D_1=\{(x,y):x<1,y<0\}$; $D_2=\{(x,y):x<1,y>0\}$; $D_3=\{(x,y):x>1,y<0\}$; and $D_4=\{(x,y):x>1,y>0\}$. Upon evaluation at a random point in each D_i, the ϕ-tables given in Table 6 are constructed. On each D_i, we find that each entry (operator, numerical values of arguments) in the ϕ-table for F matches an entry in the ϕ-table for f and vice versa. Also $f(X,Y)=F(X,Y)$. Thus, we conclude $f\equiv F$ on each region.

Example 6--Resolvable Case where $f\not\equiv F$. Let f(x)=abs(x) and F(x)=(x+abs(x))/2. An analysis of f yields two regions $D_1=\{x:x<0\}$ and $D_2=\{x:x>0\}$. Upon evaluation, the entries in the ϕ-tables match but $f(X_1)\neq F(X_1)$ for X_1 in D_1. The conclusion is $f\not\equiv F$.

Example 7--Unresolvable Case where $f\equiv F$. Let f(x)=exp((x-1)/2) and F(x)=(exp(x-1))$\uparrow\frac{1}{2}$. Since f is a T_1-function, only one region $D_1=\{x:-\infty<x<\infty\}$ is considered. Upon evaluation at X_1 in D_1, $f(X_1)=F(X_1)$. Since the ϕ-tables do not match, no firm decision is made

Example 8--Unresolvable Case where $f\not\equiv F$. Let f(x)=abs(x) and F(x)=abs(x)abs(x+10**10)/(x+10$\uparrow$10). The regions for investigation determined by f are $D_1=\{x:x<0\}$ and $D_2=\{x:x>0\}$. Since the entries in the ϕ-tables do not match, f and F are not resolvable. No firm decision is made unless we are fortunate enough to choose $X_1 < -10^{10}$ in D_1.

Table 6. φ-Tables for Examples 5-8

Example	φ-table for F	φ-table for f
5	abs,X-1 ln,Y^2	abs,X-1 ln, Y*Y ln,(Y-1)(Y+1)+1
6	abs,X	abs,X
7	$\uparrow$,exp(X-1),$\frac{1}{2}$	empty
8	abs,X+10^{10} abs,X	abs,X

The Subroutine MATCH

As a straight forward implementation of the method described in the previous section, a supporting set of arithmetic routines was installed in PICLS. The subroutine MATCH may be called from the PICLS language in order to numerically test f and F at a randomly selected point. MATCH is called by three operation codes: CN-correct numeric, WN-wrong numeric, and AN-anticipated numeric. The programmer would normally use these instructions to process a student's mathematical response. Executions of CN, WN, or AN cause a transfer of control to MATCH and the string of symbols following the operation code is passed as an argument to MATCH. The format of these operations is

:CN:k,f,n,V_1,L_1,R_1,...,V_n,L_n,R_n:S(RIGHT)F(WRONG)

where the string of symbols following the second colon and preceding the last is the argument. The items in the string separated by a comma have the following meaning:

f is the correct expression specified by the programmer.

n is an arithmetic expression, the value of which denotes the number of ordered triples V_i, L_i, R_i in the string.

V_i is the name of a (simple, singly-doubly subscripted) variable.

L_i and R_i are arithmetic expressions whose values denote the real interval $[L_i, R_i]$ from which a random number is selected and assigned to V_i.

k is an instruction flag which may assume the values ±1,±2,±3,±4. If repeated evaluations are needed, one can take advantage of the fact that f and/or F have been compiled and are in a form for rapid evaluation.

If $|k|=1$, use the new F and the previous f, ignoring any specified f in the argument string. If $|k|=2$, use the previous F and the new f. If $|k|=3$, use the new F and the new f. If $|k|=4$, use the previous F and the previous f. If $k>0$ and f and F are not resolvable due to Φ-operators, yet $f(X)=F(X)$ for each random X, print "LOOK OK. YOUR ANSWER SHOULD REDUCE TO f" where f is the expression extracted from the argument string. If $k<0$, suppress the printing of the above message.

Upon a call to MATCH, the following activities take place.

(1) Evaluate k.

(2) If $|k|=2$ or 3, compile f as specified in the argument string and place the code in the correct answer array for later evaluation. If $|k|=1$ or 4, ignore the f in the argument string and assume the previously compiled f, currently residing in the correct answer array.

(3) If $|k|=1$ or 3, fetch F from the student buffer and compile

the expression. If compilation is successful, place the result in the student answer array. If a syntax error is found, print the appropriate error message and exit from MATCH. This exit is not the normal failure exit in that the answer is not registered as incorrect, but rather as one which has no meaning. The exit is to the point where the student can type a new F. If $|k|=2$ or 4, assume the F which already resides in compiled form in the student answer array.

(4) Evaluate n.

(5) If $n\leq0$, ignore this step. If $n>0$, then for $i=1,\ldots,n$, generate a random number in the interval $[L_i,R_i]$ and store it in the location for V_i.

(6) Evaluate f and F using the random values for the V_i. If a ϕ-operator is encountered with argument h, enter the information in the appropriate ϕ-table. If the expression for h contains no variable, no entry is made since $\phi(h)$ is constant. If ϕ=abs, then $|h|$ is entered as the argument. If ϕ denotes exponentiation to a fractional or variable power with variable base, then h consists of the double entry (base, $|\text{power}|$) where negative powers are changed to positive to allow resolvability of $g(x)^r$ and $g(x)^{-r}$.

(7) For $\delta=.5(10^{-10})$, test for any one of three conditions: $|f|<\delta$ and $|F|<\delta$; $|f-F|<\delta$; or $|(f-F)/f|<\delta$. If any are satisfied, go to Step 8. Otherwise, take the FAILURE exit.

(8) If both ϕ-tables are empty, conclude $f\equiv F$ and take the SUCCESS exit from MATCH. If only one of the ϕ-tables is empty, go to

Step 10. If neither ϕ-table is empty, go to Step 9.

(9) For each entry (ordered pair or triple) in the ϕ-table for f, search for an identical entry in the ϕ-table for F. The arguments must agree within an error tolerance of δ in the manner specified in Step 7. If, upon completion of the search, every entry in each table has been successfully matched with an entry in the other table, conclude that f and F are resolvable and take the SUCCESS exit from MATCH. If any entry, in either table, is not accounted for, go to Step 10.

(10) If $k>0$, print the conditional success message "LOOKS OK. YOUR ANSWER SHOULD REDUCE TO f". If $k<0$, suppress printing, but set a flag for future checks. In either case, take the SUCCESS exit from MATCH.

Examples 9-13 are presented below to illustrate how the programmer may typically use MATCH to check a student's answer. The programmer specifies f and determines the regions D_i from which points should be randomly selected. For the purpose of discussion, an F is also specified for the examples. Table 7 presents the corresponding ϕ-tables which are constructed by MATCH.

<u>Example 9--Two T_1-Functions.</u> Suppose the programmer specifies f(x)=x-sin(x)/(2cos(x)) as the correct answer. The region for consideration is $D_1= (-\infty,\infty)$ and a typical call to MATCH is

```
:CN:3,x-sin(x)/(2*cos(x)),1,x,-9,9:S(RIGHT)F(WRONG).
```

For any expression F specified by the student, a random X in the interval [-9,9] is selected and the resulting values of f and F are compared.

Suppose the student specifies (2*x*cos(x)-sin(x))/(2*cos(x)). Since the ϕ-tables are empty and $f(X)=F(X)$, the conclusion is $f \equiv F$ and the next PICLS instruction labelled RIGHT is executed.

Example 10--Resolvable and $f \equiv F$. Let $f(x,y)=\sin(\log(x^2)+\text{abs}(y^2-1))$. An analysis of f yields six regions: $D_1=\{x>0, y<-1\}$; $D_2=\{x>0, -1<y<1\}$; $D_3=\{x>0, y>1\}$; $D_4=\{x<0, y<-1\}$; $D_5=\{x<0, -1<y<1\}$; and $D_6=\{x<0, y>1\}$.

A typical call to MATCH IS

```
L1:CN:-3,sin(log(x**2)+abs(y**2-1)), 2,x,0,10,y,-9,-1:S(L2)F(WRONG)
L2:CN:-4,0,1,y,-1,1:S(L3)F(WRONG)
L3:CN:-4,0,1,y,1,9:S(L4)F(WRONG)
L4:CN:-4,0,1,x,-10,0:S(L5)F(WRONG)
L5:CN:-4,0,1,y,-1,1:S(L6)F(WRONG)
L6:CN:4,0,1,y,-10,-1:S(RIGHT)F(WRONG)
```

The execution of this sequence calls for a comparison of f and F in the regions D_1, D_2, D_3, D_6, D_5, and D_4. Suppose the student specifies

```
sin(log(x*x))*cos(abs((y-1)*(y+1)))+
cos(log(x**3/x))*sin(abs((y-1)**2+2*y-2)).
```

Since $f(X,Y)=F(X,Y)$ in each D_i, the failure exit to WRONG should not occur. Instead, the success exits to L2, L3,....,L6, RIGHT will be taken. In statement L1, $|k|=3$ which tells MATCH to use the f specified in the argument string and the F from the student buffer. In L2-L6, $|k|=4$, which tells MATCH to use the f and F which already exist in compiled form. In L1-L5, $k<0$ which tells MATCH to suppress the unresolvability print. In L6, $k>0$ which tells MATCH to print the unresolvability message if the condition occurred in any of the CN's, L1-L6. In this example, f and F are resolvable.

Example 11. Consider the $f(x,y)$ in Example 10 and suppose the student specifies

```
sin(log(x*x))*cos(abs((y-1)*(y+1)))
      +cos(log(x**2))*sin(y**2-1).
```

Here $f \equiv F$ on D_1, D_3, D_4, and D_6, but not on D_2 and D_5. The second CN in Example 9 would detect the condition $f(X,Y) \neq F(X,Y)$ on D_2 and the failure exit to WRONG would be taken. Resolvability is not checked on D_2 since F is unconditionally wrong.

Example 12--$f \equiv F$ but Unresolvable. Let F(x)=(x**4)**.25 and f(x)=abs(x). An analysis of f yields the regions $\{x>0\}$ and $\{x<0\}$. On both $f \equiv F$. A typical call to MATCH would be

```
  :CN:-3,abs(x),1,x,-10,0:S(L1)F(WRONG)
L1:CN:4,0,1,x,0,10:S(RIGHT)F(WRONG).
```

In the last CN, prior to an exit to RIGHT, the program prints the conditional success message "LOOKS OK. YOUR ANSWER SHOULD REDUCE TO ABS(X).".

Example 13--$f \not\equiv F$ and Unresolvable. Let F(x)=exp(ln(x)) and f(x)=ln(exp(x)). An analysis of f yields $\{-\infty,\infty\}$ as the single region. A typical call to MATCH is

```
:CN:3,ln(exp(x)),1,x,-5,5:S(RIGHT)F(WRONG).
```

If X is randomly chosen nonpositive, evaluation of F will break down and MATCH will exit to WRONG: If $X>0$, MATCH will print the conditional success message and exit to RIGHT.

The argument string which is passed to MATCH is processed from left to right which allows for random assignment of values to be functionally dependent on previously assigned values. For example, suppose we define f(x,y)=ln(abs(abs(x)-y)). The four regions are $D_1=\{x<0, -\infty<y<-x\}$; $D_2=\{x>0, -\infty<y<x\}$; $D_3=\{x<0,-x<y<\infty\}$; and $D_4=\{x>0,x<y<\infty\}$. A typical call to MATCH to test in each region would be:

```
 :CN:-3,ln(abs(abs(x)-y)),2,x,-9,0,y,-9,-x:S(A)F(W)
A:CN:-4,0,2,x,0,9,y,-9,x:S(B)F(W)
B:CN:-4,0,2,x,-9,0,y,-x,9:S(C)F(W)
C:CN:4,0,2,x,0,9,y,x,9:S(RIGHT)F(W)
```

Table 7. Φ-Tables for Examples 9-13

Example	Φ-table for F	Φ-table for f
9	empty	empty
10	log,X*X abs,(Y-1)*(Y+1) log,X**3/X abs,(Y-1)**2+2*Y-2	log,X**2 abs,Y**2-1
11	log,X*X abs,(Y-1)*(Y+1) log,X**2	log,X**2 abs,Y**2-1
12	**,X**4,.25	abs,X
13	ln,X	ln,exp(X)

Mathematical Justification of the Method

The small letters z and x will denote variables over C^n and R^n respectively. The capital letters Z and X will denote randomly selected values of z and x respectively. f and F are considered equivalent over a set S if, for each point p is S, either f(p)=F(p) or both are undefined.

Theorem 1 [18]. Let g be holomorphic in the domain D and suppose $g\not\equiv 0$. Then the set $V=\{z\in D:g(z)=0\}$ has 2n-dimensional Lebesgue measure zero.

__Definition 1 [18]__. Let D be a domain in C^n. A subset $V\subset D$ is said to be thin if for every point z in D, there are an open polydisc $\delta(z;r)\subset D$ and a function g holomorphic and not identically zero in $\delta(z;r)$ such that g vanishes identically on $V\cap\delta(z;r)$.

__Remark 1 [18]__. The set where a nonzero holomorphic function vanishes is closed, has no interior, and is thin.

__Theorem 2 [18]__. Let V be a thin subset of the connected, open subset $D\subset C^n$. Then D-V is connected.

__Theorem 3 [3]__. Let g(z) be analytic in a domain $D\subset R^{2k}$ and, for some point $(z_1^0,\ldots,z_k^0)\epsilon D$ where $z_j^0=x_j^0+iy_j^0$, let g(z) vanish in the k-dimensional rectangle $|x_j-x_j^0|<r_j, y_j=y_j^0$ for $j=1,\ldots,k$. Then g(z) vanishes in D.

We can now specify properties of g which will place a theoretical reliability on the method of investigating numerical values of g at randomly selected points. It is possible to generalize the class of T_1-functions to a larger class T*. $g:C^n\rightarrow C^1$ is in T* if g is analytic on a region (nonempty, open, connected set) D and analytic in the real sense on $D\cap R^n$ with properties (1) D is dense in C^n, (2) if L is a nonempty, open, connected set, so is $L\cap D$, and (3) if m_n denotes the n-dimensional Lebesgue measure, then $m_{2n}(C^n-D)=m_n((C^n-D)\cap R^n)=0$. Properties (1) and (3) serve to insure that a randomly selected value will fall outside the region of analyticity with probability zero. Property (2) serves to eliminate those functions with ϕ-operators. In particular, it rules out branch lines. The class T* has some closure properties. If g_1 and g_2 are in T*, then $g_1\pm g_2$, g_1g_2, and g_1/g_2 $(g_2\neq 0)$ are in T*. By verifying properties (1), (2), and (3) for

$D=D_1 \cap D_2$, where D_1 and D_2 are the domains of g_1 and g_2, it follows that sums and products are in T*. For the quotient, let $V=\{z:g_2(z)=0\}$. By Remark 1, V is thin and closed relative to D_2, so D_2-V is open, dense in C^n and, by Theorem 2, also connected. Also, $L\cap(D_2-V)$ is open, non-empty, and by Theorem 2, connected. By Theorem 1 and property (3), $m_{2n}\{C^n-(D_2-V)\}=0$. By Theorem 3 and property (3), $m_n\{(C^n-(D_2-V))\cap R^n\}=0$. So $1/g_2 \epsilon T^*$ and by the product established above, $g_1/g_2 \epsilon T^*$.

Starting with polynomials and the exponential function, it is possible to build the class the T_1-functions described in an earlier section. Given f and F in T_1 where $f \neq F$, then the two can agree only on a nowhere dense set of 2n-dimensional Lebesgue measure zero. By Theorem 3, they cannot agree on an open subset of R^n. Using the ratio of Lebesgue measures as the probability, there is a 0-probability of selecting $X \epsilon R^n$ or $Z \epsilon C^n$ where f and F have the same value.

Concluding Remarks

The discussion in this chapter was intended to display both the power and the dangers in numerically comparing the student's answer with the correct answer. The use of this matching technique in the initial experiment in teaching computational mathematics has been totally successful except for rare instances when the method failed to yield a decision because of unresolvability. However, it was also evident that the student tends to construct responses which are closely related to the instructional material. For example, if the correct answer is x, then the student is not likely to arbitrarily add and subtract the hyperbolic cosine of x. This tendency of the student along with the author's ability to analyze the correct answer lends to the method a stability which might not be realized in other applications. If processing time is no factor and an exact algebraic algorithm can be applied, it should clearly be used since the finite precision of a computer can cause failure of a numerical method. Although available methods of testing the equivalence of expressions appear to be sophisticated enough for instructional application in elementary methematics, extensions are needed for those areas in which ϕ-type operators are frequently used.

In addition to the theoretical problems caused by ϕ-operators, other problems are introduced by variables which assume only integer values. In the more general case, it is desirable to compare two expressions $f(x_{\xi_1},\ldots,x_{\xi_m})$ with $F(x_{\rho_1},\ldots,x_{\rho_m})$ where x is a vector and ξ_i and ρ_i are integer-valued subscript expressions depending on integer-valued variables. For the purposes of testing equivalence of f and F, we treat the members of the array x as independent real variables. Difficulty arises in uniquely identifying a member of the array by considering the associated subscript expression. Since we have a mapping of integers into integers, random sampling can easily yield the wrong conclusion. For example, let $f(x_k)=x_k$, $F(x_k=x_{(k^2/2-k/2+1)}$. Then $f(x_k)=F(x_k)$ for $k=1$ or 2 and $f(x_k)\neq F(x_k)$ elsewhere. Also, integer-valued variables may occur in the expressions as nonsubscripts, e.g. $f(x_k)=k\cdot x_k^{k-1}$. As a programming technique in the development of the computational mathematics course, a random value is generated for each subscript variable. The values of the subscript expressions are rounded to the nearest integer and then reduced modulo the dimension of the array in order to identify a position in the array. The expressions are then numerically compared as before. This process is then repeated with values of the subscript variables increased by one. If both numerical comparisons succeed, the expressions are assumed to be equivalent. In an instructional environment, this method has been totally successful in spite of its obvious defects.

A major effort is needed in the areas of structure and content analysis. A student's answer may be correct from the standpoint of equivalence but not in a form for economical evaluation. Nesting of polynomials and forming the sum of numbers starting with the smallest and ending with the largest are two simple examples where it might be useful if the structure of the student's answer could be analyzed. If the student's answer is incorrect, a content analysis is needed to determine how it differs from the correct answer. Manacher [27] proposes using sequences of numbers to check for such properties as symmetry, correct boundary conditions, and linearity of variables. General advances in content analysis would be a step toward detecting the source of the student's error.

CHAPTER III

DESIGN AND DEVELOPMENT OF THE CAI COURSE

General Philosophy and Design Considerations

A variety of factors must be considered in the design and construction of CAI course materials. While a clear specification of the course objectives is necessary, one must also consider the capabilities of the available hardware and software and the current practices in instructional design. If the course is to be of significant duration, the element of time may impose additional constraints on the sophistication of the end product. In particular, a large expenditure of man hours is required to develop extensive remedial sequences and multitrack programs. If the effectiveness of the program is to be tested in a production environment, the materials must be organized for ease in administration. This section is a discussion of how these factors affected the design and construction of the CAI course in computational mathematics.

As defined by this author, the purpose of a course in computational mathematics is to teach the student how to analyze mathematical problems and apply numerical methods for an approximate solution. There is a definite emphasis on problem solving. In terms of ideal student performance, the following general course objectives are stated.

1. The student should understand the theoretical developments which justify the existence of an algorithm. For a given problem, the student should determine if the theoretical

conditions are satisfied prior to applying an algorithm.

2. The student should display proficiency in the mechanics of applying an algorithm by working several standard problems.
3. Whenever applicable, the student should determine a priori bounds on the error of approximation by analyzing the error equation.
4. Whenever applicable, the student should estimate the accuracy of the solution by interpreting computational results.

A CAI course in this subject matter should attempt to remove any cumbersome arithmetic or programming requirements which might prevent achievement of these objectives.

At the outset, the course materials were paralleled with CS 414, the undergraduate numerical analysis course at Purdue University. The listed prerequisites for CS 414 were a working knowledge of a computer language and successful completion of the elementary calculus courses. The CAI course assumes the elementary calculus but programming is needed only to the extent that a student must be able to formulate mathematical expressions in a Fortran notation. The prerequisites for CS 414 have been recently upgraded to include an elementary course in linear algebra. This change is not reflected in the CAI course.

Twenty-four CAI lessons were developed for six general areas of study:

1. errors in representation of numbers and computation (1 lesson)
2. root-finding methods (10 lessons)
3. solution of linear systems (5 lessons)
4. numerical differentiation (2 lessons)

5. numerical integration (3 lessons)
6. solution of differential equations (3 lessons)

In order to concentrate on the computational and programming difficulties and, at the same time, maintain the standards of the course objectives, a typical lesson consists of three modes of instruction. They are referred to as the tutorial mode, the problem mode, and the investigation mode. These modes are designed to provide the student with increasing flexibility in the problem solving aspects of the course. Three subsequent sections of this chapter are devoted to a description of these modes.

The Student Manual presented in Appendix A was created to handle the problems of administering a CAI course in a production environment. This manual prescribes a systematic approach to the study of each lesson. By following a simple outline, the student may complete the various study activities required in a lesson and gain immediate access to any section of CAI materials. The Student Manual is intended to be self-explanatory and a further description will not be presented here.

In the area of software support, PICLS was extended to incorporate special routines needed for a more flexible course development. One such routine is the function matching program described in Chapter II. This involved a compilation subroutine which accepts arithmetic expressions from a terminal, performs a syntactic analysis, and outputs polish expressions, and an interpreter subroutine which evaluates the polish expressions. This body of special arithmetic routines served as a basis for other needed features. In those portions of the instruction where the student is expected to formulate a number of mathematical expressions, subroutines were written to store the expressions and retrieve them for evaluation at a later point in time. Once the syntax of an expression has been checked, the author-programmer may save the expression by a function call `:FJ:SAVFCT(N)` where N is an integer-valued variable. Any of the previously stored expressions can be evaluated with the numerical result stored in X by the function call `:FJ:FCTVAL(N, X)`. The specification of the location X and the function number N are under the internal control of the programmer.

The linear notation and restricted character set of the teletype terminal had a definite effect on the design and development of the course material. Special notation had to be defined and the instruction had to include a careful explanation of this notation. Examples of special notation can be found in the lessons on the Newton-Bairstow method, numerical integration, and differential equations. Combined with the restrictions placed on the student response language, the development of some sections became even more difficult. For example, the notation F'X was used for the partial derivative of F with respect to X. If the student is asked to form the total derivative of Y'=F(X,Y)=X**2+Y with respect to X, then the compiler is equipped to process 2*X+X**2+Y but syntax errors would be found in the answer F'X+F'Y*F. In this case, the instructional material must clearly request an answer in terms of X and Y. Multiple choice items were used whenever it seemed unnatural to restrict the symbols in a mathematical response. Another ill effect of the linear notation was apparent in the programming of lengthy formulas. For example, expressions such as

```
Y[K+1]=Y[K]+H*(F+H*(F'X+F'Y*F)/2
+(H**2)*(F"XX+2*F"XY*F+F'X*F'Y+F"YY*F**2+F*(F'Y)**2)/6)
```

were time-consuming to format in the program and seemed unnatural to read as teletype output. Still another restriction of the teletype terminal is its lack of graphic capabilities. The graphs and diagrams normally used in a conventional classroom were usually omitted in the CAI course. In the judgement of the author, the slow typing rate and the character orientation of the teletype terminal precluded an effective use of charts and graphs. Although it would have been possible to provide work sheet graphs to assist in the instructional process, the philosophy of the investigation was to deliberately remain computer-oriented as opposed to multimedia-oriented.

The instructional strategies used in the tutorial mode were designed on the basis of what could be done in a reasonably well-defined manner in spite of a seemingly lacking technology. The author firmly agrees with educators that a carefully planned instructional design is critical to the success of CAI and some of the recommended practices were followed. The presentation consists primarily of a linear sequence but can be readily expanded to a multilevel sequence for the express purpose of accelerating the better student and decelerating the weaker student. For each question posed to the student, the strategy is of a somewhat more sophisticated design and will be explained later in this chapter. The following reasons are offered for not designing and implementing a highly sophisticated instructional strategy for the initial system.

1. Man hour requirements could be expected to increase at least linearly with the number of tracks.
2. Experience was needed to establish that the software-hardware

complex was a workable system.

3. Experience was needed to determine the general reaction of students to CAI for this level and type of mathematical material in order to establish a basis for easier and more difficult tracks.

The next three sections of this chapter describe the purposes and structure of the tutorial mode, the problem mode, and the investigation mode as they exist in the current system. Excerpts of course material are presented to demonstrate particular concepts. No attempt is made in this chapter to describe the subject content of the entire body of course material in computational mathematics. A general description of the CAI course material in each of the twenty-five lessons is presented in Appendix B. For an appreciation of the depth of the student involvement in each of the three modes, the reader is referred to the sample teletype output in Appendix D.

Structure of the Tutorial Mode

The tutorial mode is designed for each lesson with the traditional classroom in mind. Its purpose is to provide the student with those instructional experiences which would be feasible in the classroom if sufficient time and resources were available. In keeping with the course objectives, the following activities are typical in this mode of instruction.

1. The student is led through the theoretical concepts surrounding a particular method. The student actively participates through constructed responses to questions or multiple choice items.

2. The student participates in a variety of examples and exercises which demonstrate particular concepts and which are interspersed at appropriate places throughout the theoretical developments.
3. The student is led through the analysis of a typical problem and supplies the mathematical formulas needed to apply the algorithm.
4. The student concentrates on the development and formulation and is free from cumbersome arithmetic. This is basically accomplished by allowing the student to construct responses which are equivalent to the correct answer and left in unreduced form.

Prior to beginning the tutorial mode, the student is expected to complete an outside reading assignment. Since the instruction is designed for the average student, the faster student may find this mode to be a review of the outside reading assignment while the slower student is expected to experience greater difficulty and benefit more from the material. All students are exposed to the same core material since the presentation is basically a linear sequence. A skeleton strategy for individualization is incorporated at the item level. At this level, the slower student is momentarily detained and, hopefully, his difficulty will be remedied. The individualization strategy for multiple choice items and constructed mathematical response items are shown in Figures 1 and 2. Multiple choice items are handled in a somewhat simplified manner since the student must select one of a predetermined set of possible answers. Due to a lack of knowledge of how students respond, the strategy for handling constructed mathematical responses is more complicated and the anticipation of incorrect answers is a difficult task. In order to offset the lack of anticipated answers, the student may at any time type HELP and additional information or hints will be provided. If a student types two successive unanticipated answers, the normal procedure is to give him the correct answer along with a detailed explanation.

The following example of PICLS code involving a constructed mathematical response is taken from the tutorial mode of Lesson 2. This code demonstrates the instructional strategy depicted in Figure 2.

```
L12:TY:ON I=[0,2], WRITE AN EXPRESSION FOR MAX(ABS(G'(X))) BY
   :TY:CHOOSING A PARTICULAR VALUE FOR X FROM I=[0,2].
Q12:QU:MAX(ABS(G'(X)))=
   :AA:HELP:S(Q12)
   :TY:     G'(X)=-EXP((1-X)/2)/2.  G"(X)=EXP((1-X)/2)/4.  SINCE THE EXP
   :TY:     FUNCTION IS NEVER 0, G"(X) IS NEVER ZERO, THAT IS, G'(X) HAS
   :TY:     NO RELATIVE EXTREME POINTS.  HENCE, THE MAXIMUM ON I=[0,2]
   :TY:     MUST OCCUR AT ONE OF THE ENDPOINTS.  TRY AGAIN.
   :CN:3,EXP(1/2)/2,0:S(L13)
   :TY:OK
   :WN:-3,EXP(-1/2)/2,0:S(Q12)
   :TY:     NO.  YOU USED THE WRONG ENDPOINT OF I=[0,2].  TRY AGAIN.
   :WN:-3,-EXP(-1/2)/2,0:S(Q12)
   :TY:     NO.  YOU USED THE WRONG ENDPOINT OF I=[0,2].  ALSO, THE
   :TY:     ABSOLUTE VALUE SHOULD MAKE YOUR ANSWER POSITIVE.  TRY AGAIN.
   :WN:-3,-EXP(1/2)/2,0:S(Q12)
   :TY:     NO.  THE ABSOLUTE VALUE SHOULD MAKE YOUR ANSWER POSITIVE.
   :TY:     TRY AGAIN OR TYPE HELP.
   :UN:     NO.  TRY AGAIN OR TYPE HELP.
   :TY:MAX(ABS(G'(X)))=
   :NO:
   :TY:     NO.  MAX(ABS(G'(X))) ON [0,2] OCCURS AT X=0.  THE ANSWER IS
   :TY:     MAX(ABS(G'(X)))=EXP(1/2)/2.
   :RD:PRESS (RETURN) TO CONTINUE.
L13:TY:SO ABS(G'(X))< 1 ON I=[0,2].  SINCE ALL CONDITIONS OF THE LINEAR
```

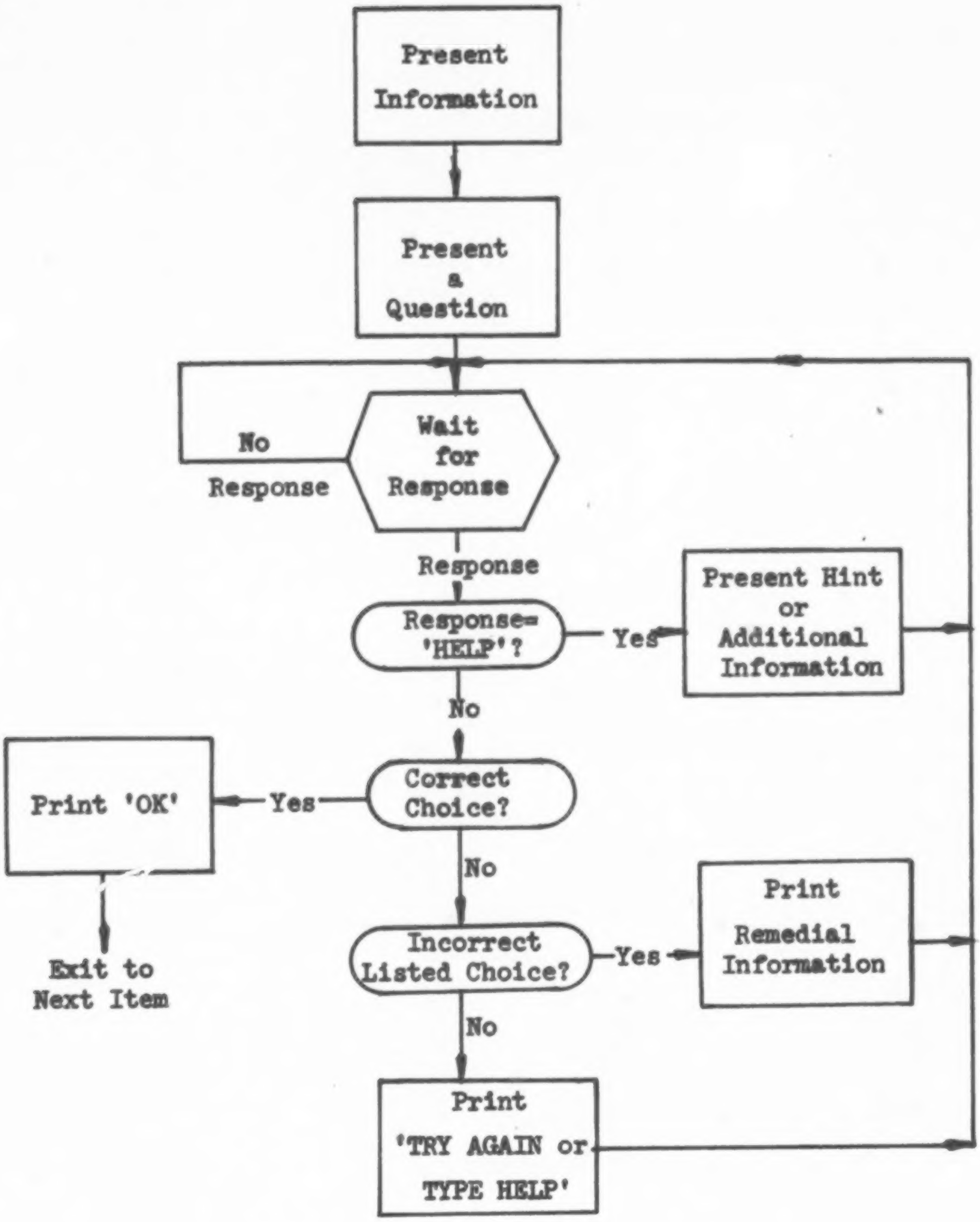

Present Information
Present a Question
Wait for Response
No Response
Response
Response= 'HELP'?
Yes
Present Hint or Additional Information
No
Correct Choice?
Yes
Print 'OK'
Exit to Next Item
No
Incorrect Listed Choice?
Yes
Print Remedial Information
No
Print 'TRY AGAIN or TYPE HELP'

Figure 1. Instructional Strategy for Multiple Choice Items

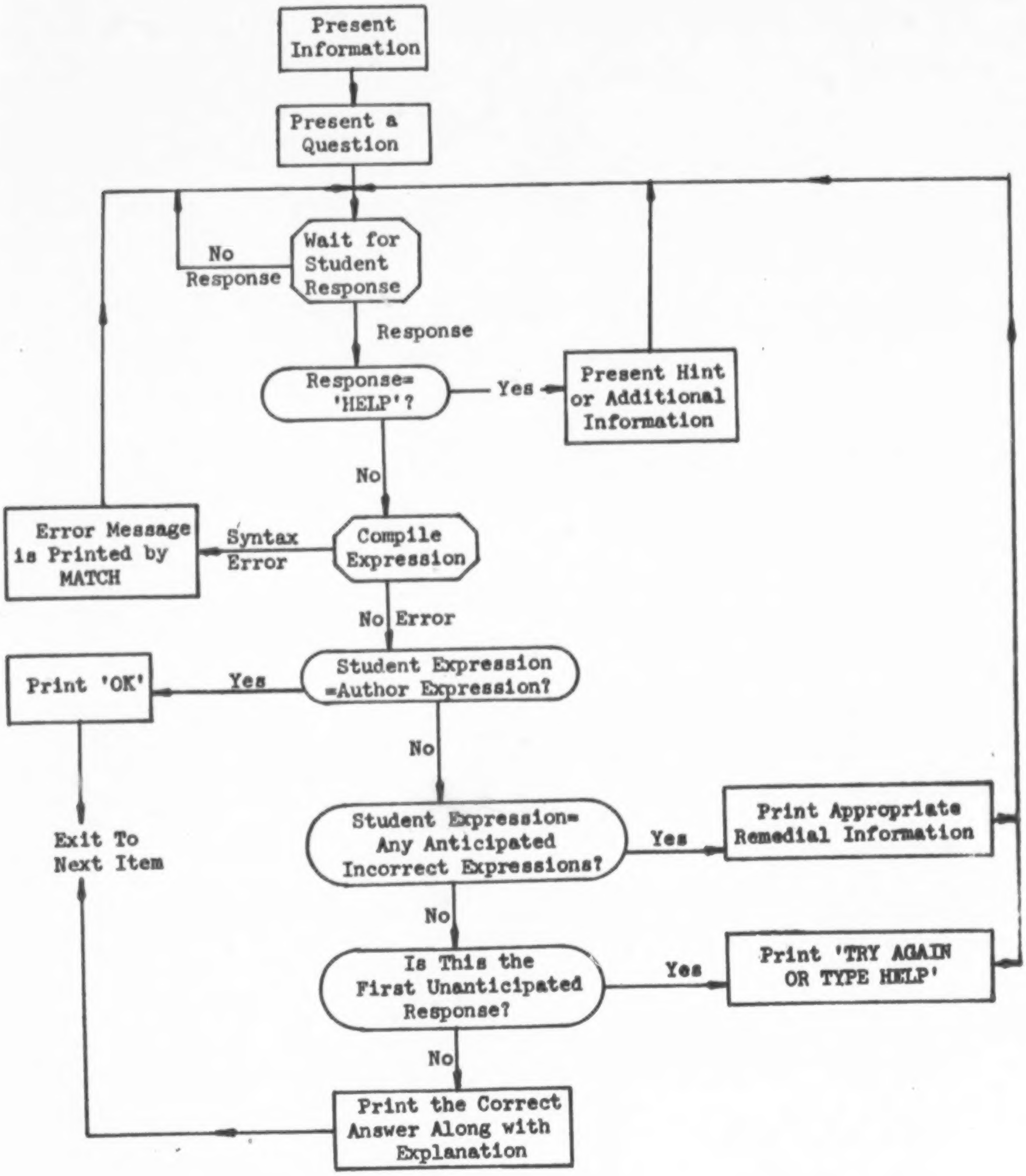

Present Information
Present a Question
Wait for Student Response
No Response
Response
Response= 'HELP'?
Yes
Present Hint or Additional Information
No
Compile Expression
Syntax Error
Error Message is Printed by MATCH
No Error
Student Expression =Author Expression?
Yes
Print 'OK'
Exit To Next Item
No
Student Expression= Any Anticipated Incorrect Expressions?
Yes
Print Appropriate Remedial Information
No
Is This the First Unanticipated Response?
Yes
Print 'TRY AGAIN OR TYPE HELP'
No
Print the Correct Answer Along with Explanation

Figure 2. Instructional Strategy for Constructed Mathematical Responses

Prior to the execution of this block of instruction, the student has derived the iteration function G(X)=EXP((1-X)/2). In this block of instruction, the student is to ascertain that | G'(X) | < 1 on the interval [0,2] by actually computing max | G'(X) | . The correct answer is specified as EXP(1/2)/2 while EXP(-1/2)/2, -EXP(-1/2)/2, and -EXP(1/2)/2 are anticipated incorrect responses. If the student answers correctly, he advances to the new material beginning at label L13. If the student gives two successive unanticipated answers, he is given the information following the operation :NO:. He then begins the new material by pressing the Return Key. If the student enters a syntactically incorrect expression, the subroutine MATCH (see Chapter II) prints an appropriate error message. Depending on the student, the twenty-five PICLS instructions listed above can create several variations of teletype output. The following dialogue between the student and the program illustrates one possibility. At those points where a student must respond, PICLS types a # sign at the left margin.

```
 ON I=[0,2], WRITE AN EXPRESSION FOR MAX(ABS(G'(X))) BY
 CHOOSING A PARTICULAR VALUE FOR X FROM I=[0,2].
 MAX(ABS(G'(X)))=
#1/2
      NO.  TRY AGAIN OR TYPE HELP.
 MAX(ABS(G'(X)))=
#.5*EXP(-.5)
      NO.  YOU USED THE WRONG ENDPOINT OF I=[0,2].  TRY AGAIN.
 MAX(ABS(G'(X)))=
#.5*EXP(1)**.5
 OK
 SO ABS(G'(X))<1 ON I=[0,2].  SINCE ALL CONDITIONS OF THE LINEAR
```

Another possibility is illustrated by the following dialogue.

```
 MAX(ABS(G'(X)))=
# HELP
    G'(X)=-EXP((1-X)/2)/2.  G"(X)=EXP((1-X)/2)/4.  SINCE THE EXP
    FUNCTION IS NEVER 0 , G"(X) IS NEVER ZERO, THAT IS, G'(X) HAS
    NO RELATIVE EXTREME POINTS.  HENCE, THE MAXIMUM ON I=[0,2]
    MUST OCCUR AT ONE OF THE ENDPOINTS.  TRY AGAIN.
 MAX(ABS(G'(X)))=
# 0

    NO.  TRY AGAIN OR TYPE HELP.
 MAX(ABS(G'(X)))=
# -EXP(1)/2
    NO.  MAX(ABS(G'(X))) ON [0,2] OCCURS AT X=0.  THE ANSWER IS
    MAX(ABS(G'(X)))=EXP(1/2)/2.
 PRESS (RETURN) TO CONTINUE.
#
```

A third possibility which also illustrates a syntax error is the following dialogue.

```
 MAX(ABS(G'(X)))=
# EXP(.5(/2
 ILLEGAL CHARACTER OR COMBINATION      5(
 TYPE A CORRECT EXPRESSION
# EXP(.5)/2
 OK
```

The reader is referred to Appendix D for the teletype output of a complete tutorial mode. Unlike the problem and investigation modes, the process of instruction in the tutorial mode is under the direction of the computer program.

Structure of the Problem Mode

The problem mode is designed to provide the student with the instructional experience derived from solving several typical problems. In keeping with the objectives of the course, the student is required to

1. analyze the problems and construct the necessary formulas for

application of an algorithm,

2. input his formulas and define values for any parameters associated with the algorithm, and
3. direct the computer to a numerical solution.

A major characteristic of this mode is the complete freedom from bookkeeping chores normally associated with programming. Once the student has correctly formulated the necessary equations, the computer assumes the bookkeeping and computational work. If the computation is open-ended (e.g. iterative methods or extrapolation to the limit), the student is provided with one logical step of the computational results each time he pushes the Return Key. The student terminates this type of problem by typing STOP. If the computation is dependent on parameters supplied by the student (e.g. initial estimates for iterative methods or the step-size for numerical differentiation, integration, and the solution of differential equations), the student always has the option of redefining the parameters and repeating the calculation without retyping the equations. Thus, a problem may be easily reworked in several ways.

Unlike the tutorial mode, the problem mode does not assign an active teaching function to the computer. Instead, it calls for specific formulas needed to apply an algorithm to a problem and the student must display his ability to work computational problems by supplying the correct formulas. Except for isolated places, the student cannot call for HELP. In the event of an incorrect answer, remedial material is practically nonexistent. Where it does exist, it appears as a statement of fact and is not intended to remedy a misunderstanding of concepts. Thus, the student must either supply the correct equations or terminate the problem. If the student must terminate a problem, he is expected to review his output from the tutorial mode in order to remedy his difficulty. This overall philosophy is employed in an attempt to establish independence of outside help. Except for YES/NO options made available to the student for reformulating a problem, the problem mode consists entirely of constructed responses. The typical strategy for processing a single response is shown in Figure 3.

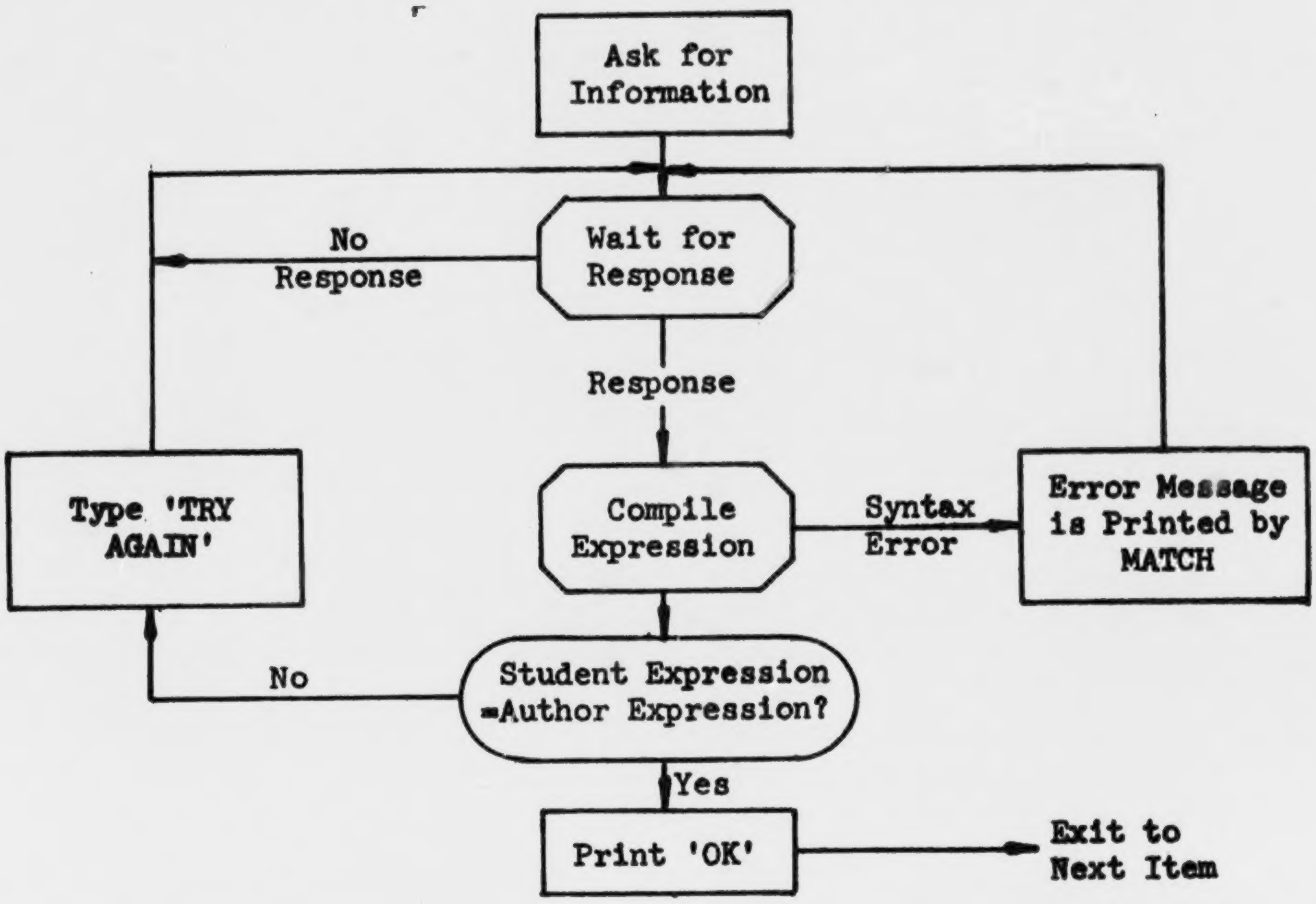

Ask for Information
Wait for Response
No Response
Response
Type 'TRY AGAIN'
Compile Expression
Syntax Error
Error Message is Printed by MATCH
No
Student Expression =Author Expression?
Yes
Print 'OK'
Exit to Next Item

Figure 3. Problem Mode Strategy for Constructed Responses

2

The interested reader may consult Appendix D for the teletype output of a complete problem mode. The following example of PICLS code deals with the trapezoidal rule and is taken from the problem mode of Lesson 18.

```
PROB1:TY:LET F(X)=SQRT(X)+1/SQRT(X).  WE WISH TO APPROXIMATE
:TY:INTEGRAL(F(X);[1,2])
:TY:SPECIFY THE ERROR IN TERMS OF H AND Z.
:ST:A=1
:ST:B=2
P1:QU:E(H)=
:AA:STOP
R1:TY:SELECT ANOTHER PROBLEM.:(Q1)
:CN:3,(-H↑2)*(3-Z)/(SQRT(Z↑5)*48),2,H,-3,3,Z,0,1.5:S(P2)
:TY:OK.  E(H)=(-H↑2)*(3-Z)/(48*Z↑2.5)
:UN:     TRY AGAIN.  F'(X)=.5*(X↑(-1/2)-X↑(-3/2)).
:TY:E(H)=
:UN:     TRY AGAIN.  F"(X)=.25*(3-X)/(X↑2.5).
:TY:E(H)=
:UN:     TRY AGAIN.:(P1)
:NO:
P2:TY:ANALYTICALLY DETERMINE AN H SO MAX(ABS(E(H))<.5*10↑(-2) ON [1,2].
:TY:DO THIS BY USING MAX(3-X), MIN(48*Z↑2.5) ON [1,2].
P3:QU:H=
:AA:STOP:S(R1)
:CN:3,H,1,H,-9,9:(P4)
P4:ST:H=ANSWER
:IF:H 'LT' 0:S(P3)
:TY:     H MUST BE POSITIVE.
:IF:H  'GT' .2*SQRT(3)+.02:S(P3)
:TY:     YOUR CHOICE OF H IS TOO LARGE.
:IF:H 'LT' .2*SQRT(3)-.02:S(P3)
:TY:     YOUR CHOICE OF H IS TOO SMALL.
:NO:
:ST:H=.2*SQRT(3)
:FJ:OUTPUT(1,H,(29H OK.  ACTUALLY, H=.2*SQRT(3)=,E23.15))
:TY:THIS H YIELDS THE NUMBER OF SUBDIVISIONS
P6:QU:N=
:AA:STOP:S(R1)
:CN:3,(B-A)/H,0:S(P7)
:CN:-3,INT((B-A)/H),0:S(P7)
:CN:-3,INT((B-A)/H)+1,0:S(W8)
:ST:N=3
:UN:     NO.  THE SPACING OF POINTS FOR THE TRAPEZOIDAL RULE
:TY:     IS ALWAYS H=(B-A)/N.  N, OF COURSE, IS CHOSEN TO BE AN
:TY:     INTEGER.:(P6)
:NO:
```

```
P7:ST:N=INT((B-A)/H)+1
:FJ:OUTPUT(1,N,(38H WE CHOOSE THE FIRST LARGER INTEGER N=,F2.0))
W8:CN:3,Z,1,Z,-9,9:(P8)
P8:NO:
:TY:WRITE THE H AND THE TRAPEZOIDAL RULE FOR N SUBDIVISIONS.
P17:QU:H=
:AA:STOP:S(R1)
:CN:3,(B-A)/N,0:S(P18)
:UN:     NO.  TRY AGAIN.:(P17)
P18:NO:
:ST:H=(B-A)/N
:ST:F[0]=2
:ST:F[1]=3.5/SQRT(3)
:ST:F[2]=8/SQRT(15)
:ST:F[3]=3/SQRT(2)
:ST:TT=(F[0]+2*(F[1]+F[2])+F[3])/6
P19:QU:IT=
:AA:STOP:S(R1)
:CN:3,TT,0:S(P20)
:UN:     TRAPEZOIDAL RULE WITH FOUR POINTS SINCE N+1=4.:(P19)
P20:FJ:OUTPUT(1,TT,(8H OK. IT=,E23.15))
:ST:TT=2*(5*SQRT(2)-4)/3
:FJ:OUTPUT(1,TT,(40H THE TRUE VALUE IS INTEGRAL(F(X);[1,2])=,E23.15))
```

The following teletype output represents one possible successful path through the above code.

```
 LET F(X)=SQRT(X)+1/SQRT(X).  WE WISH TO APPROXIMATE
 INTEGRAL(F(X);[1,2])
 SPECIFY THE ERROR IN TERMS OF H AND Z.
 E(H)=
#-(H↑2)*(3-Z)/(SQRT(Z↑5)*48)
 OK.  E(H)=(-H↑2)*(3-Z)/(48*Z↑2.5)
 ANALYTICALLY DETERMINE AN H SO MAX(ABS(E(H)))<.5*10↑(-2) ON [1,2].
 DO THIS BY USING MAX(3-X), MIN(48*Z↑2.5) ON [1,2].
 H=
#(3/25)**.5
 OK.  ACTUALLY, H=.2*SQRT(3)=    .346410161513774E+00
 THIS H YIELDS THE NUMBER OF SUBDIVISIONS
 N=
#3
 WRITE THE H AND THE TRAPEZOIDAL RULE FOR N SUBDIVISIONS.
 H=
#1/3
 IT=
#(H/2)*(F[0]+2*(F[1]+F[2])+F[3])
 OK.  IT=    .204899241064024E+01
 THE TRUE VALUE IS INTEGRAL(F(X);[1,2])=    .204737854124363E+01
```

Prior to entering the problem mode, the student is expected to complete the tutorial mode and then consult the Student Manual for a statement of the problems. In this way, the student can leave the terminal in order to analyze and formulate the equations and return at a later time to input his formulas and obtain a numerical solution.

Structure of the Investigation Mode

The philosophy of this mode differs from the other two in that the computer does not assume an active teaching role and the student is not required to demonstrate previously acquired knowledge. It is designed to release the student from the constraints of the other modes and provide facilities for the rapid solution of problems originated by the student. Structurally, the investigation mode is similar to the problem mode. The student must formulate equations in order to apply an algorithm and, in turn, the computer assumes the usual bookkeeping chores associated with normal programming and provides numerical results. As formulas are input, they are checked only for syntax errors and saved for later evaluation. The following dialogue is a possible excerpt from the investigation mode of Lesson 7. It shows how a student may approximate the numerical solution to a system of equations.

```
 DEFINE THE ITERATION EQUATIONS
 X[K+1]=
#.1*SIN(X[K])+.2*COS(Y[K])
 Y[K+1]=
#.1*COS(X[K])-.2*SIN(Y[K])
 DEFINE THE STARTING VALUES
 X[0]=
#1/5
 Y[0]=
#0
 EACH TIME THE (RETURN) KEY IS PUSHED, TWO ITERATIONS WILL BE
 PRINTED.  TYPE 'STOP' TO TERMINATE THE ITERATION.
#
```

```
   K      X[K]            Y[K]        K       X[K]            Y[K]

   1   .21986693E+00   .98006658E-01  2   .22085021E+00   .78022681E-01
#
   3   .22129748E+00   .81982447E-01  4   .22127783E+00   .81183220E-01
#
   5   .22128894E+00   .81342965E-01  6   .22128743E+00   .81310877E-01
#
   7   .22128780E+00   .81317307E-01  8   .22128773E+00   .81316017E-01
#STOP
 DO YOU WISH TO TRY A DIFFERENT (X[0],Y[0])?
#NO
 DO YOU WISH TO REDEFINE THE ITERATION EQUATIONS?
#
```

Since the problems originate with the student, he must determine if his formulation is correct and he must interpret the numerical results.

The investigation mode is optional and may be used by the student at any time. Prior to beginning an investigation mode, the student is expected to consult the Student Manual for a format description of the required formulas. Hopefully, the tutorial and problem modes provide a source of problems for investigation. In any event, suggested problems are stated in the Student Manual. The reader is referred to Appendix D for the teletype output of a complete investigation mode.

Special Program Features

In selected problem and investigation modes, the student is given the option of using partial precision arithmetic in the computation as an aid in the study of loss of significance or the propogation of round-off error. In other places, the computer is used to generate a virtually inexhaustible supply of problems and the student has the option of requesting such a problem from the computer. For both features, the computer appears to be an ideal medium and several applications are discussed in this section.

Partial precision is optionally available to the student in the problem-investigation mode of Lesson B, in the investigation mode of Lesson 11, and in the problem-investigation mode of Lesson 13. In each of the three modes, the student may specify a precision of $p=4$, 6, or 8 decimal digits. Full single precision is assumed in Lesson B if the student specifies $p=15$. The internal effect is to round each normalized floating number in the $(p+1)$-st digit and retain the first p digits. If x is an input or the result of an arithmetic operation +,-,*,/, and x is not zero, it is reduced to a p-significant decimal digit normalized floating point number by the following algorithm:

$$m \leftarrow \log_{10} |x|$$

$$m \leftarrow \begin{cases} m+1 & \text{if } m > 0 \\ m+1 & \text{if } m \leq 0 \text{ and } m \text{ is an integer} \\ m & \text{otherwise} \end{cases}$$

$$k \leftarrow p - \mathrm{Int}(m)$$

$$x \leftarrow \mathrm{Int}\left(x \cdot 10^k + .5 \cdot \frac{|x|}{x}\right) / 10^k$$

Since the mantissa of a floating point number consists of 48 binary bits, p is restricted to the range $1 \leq p < 15$. Since the internal arithmetic is binary, the algorithm provides an approximate p-digit decimal calculator.

In the study of ill-conditioned linear systems, the following algorithm was used to generate the $n \times n$ coefficient matrix A in the problem-investigation mode of Lesson 13.

1. Select an integer i at random so that $1 < i \leq n$.
2. For each $j \neq i$ and $k=1,\ldots,n$, randomly select $a_{jk} \in (-9,9)$.

3. For k=1,...,i-1, randomly select $r_k \in (-9,9)$. If $r_k=0$ for all k, repeat this step.
4. Compute the multipliers $\alpha_p = r_p^2 / \sum_{k=1}^{i-1} r_k^2$ for p=1,...,i-1.
5. Compute the ith row as a "nearly" linear combination of the first i-1 rows by $a_{ik}=[1-(.1)^{k+1}]S_k$ for k=1,...,n where $S_k = \sum_{p=1}^{i-1} \alpha_p a_{pk}$.
6. If any row has less than two nonzero elements, restart at Step 2.

For this class of matrices, one can bound the normalized determinant by $|\text{norm}\,|A|| < .0102$. Denote the ik cofactor of A by $(-1)^{i+k}|A_{ik}|$ and define $\beta_j = (\sum_{k=1}^{n} a_{jk}^2)^{\frac{1}{2}}$. Then

$$\frac{||A_{ik}||}{\beta_1 \cdots \beta_n} \le \frac{||A_{ik}||}{\beta_i \prod_{\substack{j=1 \\ j \ne i}}^{n} (\beta_j^2 - a_{jk}^2)^{\frac{1}{2}}} = \frac{|\text{norm}\,|A_{ik}||}{\beta_i} .$$

Expanding on the ith row,

$$|A| = \sum_{k=1}^{n} (-1)^{i+k} \; a_{ik}\,|A_{ik}| = \sum_{k=1}^{n} (-1)^{i+k} \; S_k[1-(.1)^{k+1}]\,|A_{ik}|$$

$$= \sum_{k=1}^{n} (-1)^{i+k+1}(.1)^{k+1} \, S_k \,|A_{ik}|$$

and

$$| \text{norm} \, | A || \leq \sum_{k=1}^{n} (.1)^{k+1} \frac{| S_k | \; || A_{ik} ||}{\beta_1 \ldots \beta_n} \leq \frac{1}{\beta_i} \sum_{k=1}^{n} (.1)^{k+1} | S_k | \cdot | \text{norm} \, | A_{ik} ||$$

$$\leq \frac{1}{\beta_i} \sum_{k=1}^{n} (.1)^{k+1} | S_k | .$$

Using

$$\beta_i = \Big(\sum_{k=1}^{n} [1-(.1)^{k+1}]^2 S_k^2 \Big)^{\frac{1}{2}} \geq [1-(.1)^2] \Big(\sum_{k=1}^{n} S_k^2 \Big)^{\frac{1}{2}}$$

and

$$\sum_{k=1}^{n} (.1)^{k+1} | S_k | \leq \Big(\sum_{k=1}^{n} (.01)^{k+1} \sum_{k=1}^{n} S_k^2 \Big)^{\frac{1}{2}},$$

we have

$$| \text{norm} \, | A || \leq \frac{\Big(\sum_{k=1}^{n} (.01)^{k+1} \Big)^{\frac{1}{2}}}{.99} = \frac{1}{99} \left\{ \frac{1-(.01)^n}{.99} \right\}^{\frac{1}{2}} .$$

This class of matrices is conditioned to significantly perturb the true solution if the student elects to use 4-digit accuracy. Even for p=6 or 8, the concept of an ill-conditioned system is usually demonstrated. The student may observe the difference in the results by using several values for p. As an example of the above discussion, the following dialogue may take place in the problem-investigation mode of Lesson 13.

```
 PROBLEM 6.
 STATE THE DESIRED DIMENSION OF THE A-MATRIX (2,3,4,NONE).  N=
#3
 DO YOU WISH TO DEFINE YOUR OWN A-MATRIX AND B-VECTOR?
#NO
 M=(4,6,8)=NO.  SIGNIFICANT FIGURES FOR INTERNAL COMPUTATIONS.  M=
#6
 THE A-MATRIX AND B-VECTOR ARE NOW BEING SETUP --- WAIT.
 NOW READY FOR GAUSSIAN ELIMINATION
 THE CURRENT AUGMENTED MATRIX IS
   .185938E+01   .434347E+01   .563152E+01   .386653E+02
   .315264E+01  -.334289E+01   .568190E+01  -.344478E+01
   .273138E+01  -.100216E+01   .566600E+01   .945906E+01
 DO YOU WISH TO INTERCHANGE ROWS?
#YES
 SPECIFY I AND J FOR INTERCHANGE OF ITH AND JTH ROWS.
 I=
#1
 J=
#2
 THE CURRENT AUGMENTED MATRIX IS
   .315264E+01  -.334289E+01   .568190E+01  -.344478E+01
   .185938E+01   .434347E+01   .563152E+01   .386653E+02
   .273138E+01  -.100216E+01   .566600E+01   .945906E+01
 DO YOU WISH TO INTERCHANGE ROWS?
#NO
 WAIT FOR CURRENT STAGE OF GAUSSIAN ELIMINATION TO BE PERFORMED.
 THE CURRENT AUGMENTED MATRIX IS
   .315264E+01  -.334289E+01   .568190E+01  -.344478E+01
  0.             .631506E+01   .228042E+01   .406970E+02
  0.             .189405E+01   .743320E+00   .124436E+02
 DO YOU WISH TO INTERCHANGE ROWS?
#NO
 WAIT FOR CURRENT STAGE OF GAUSSIAN ELIMINATION TO BE PERFORMED.
 THE CURRENT AUGMENTED MATRIX IS
   .315264E+01  -.334289E+01   .568190E+01  -.344478E+01
  0.             .631506E+01   .228042E+01   .406970E+02
  0.            0.             .593630E-01   .237500E+00
 DO YOU WANT NORM(DET(A))?
#YES
 NORM(DET(A))=  -.345434331147931E-02
 DO YOU WANT THE SOLUTION FOR X BY BACK-SUBSTITUTION?
#YES
 X3=     .400081E+01
 X2=     .499971E+01
 X1=    -.300177E+01
 DO YOU WANT THE RESIDUALS?
#YES
           (Student directs computer to a solution of the error system)
 E3=    -.746691E-03
 E2=     .269649E-03
 E1=     .163118E-02
```

```
THE IMPROVED SOLUTION IS
X3=      .400006E+01
X2=      .499998E+01
X1=     -.300014E+01
DO YOU WANT THE RESIDUALS?
```

Another example of computer supplied problems can be found in Lesson 10 dealing with the Newton-Bairstow method. In the problem mode, third or fourth degree polynomials with random complex roots are generated for the student by the following method:

1. Randomly select α, β, A, B, and C from the interval (-9,9) for the complex root $\alpha+\beta i$ and the factor $Bx+C$ or Ax^2+Bx+C.
2. Randomly select the degree n=3 or 4.
3. If n=3, compute the coefficients a_i for the polynomial $p(x)=a_3x^3+a_2x^2+a_1x+a_0=(x^2-2\alpha x+\alpha^2+\beta^2)(Bx+C)$. If n=4, compute the coefficients a_i for the polynomial $p(x)=a_4x^4+a_3x^3+a_2x^2+a_1x+a_0=(x^2-2\alpha x+\alpha^2+\beta^2)(Ax^2+Bx+C)$.

The student is provided with the coefficients a_i and is told that p(x) has a complex root in the rectangle with vertices $(\text{Int}(\alpha)\pm 1, \text{Int}(\beta)\pm 1)$. The student must estimate a quadratic factor of p(x), define the recursion formulas for the Newton-Bairstow method, and direct the computer through successive iterations to find the quadratic factor $x^2-2\alpha x+\alpha^2+\beta^2$.

Concluding Remarks

The previous sections of this chapter are intended to describe the structure of the CAI course as it was designed and implemented. Throughout the programming and experimentation stages, it became increasingly evident that the design constraints were too stringent. An expanded or modified version of the system is needed to provide a programmer, as well as the students, with more flexibility. This section proposes some extensions or changes which can be made within the framework of existing technology.

One area which should be expanded deals with broadening the base of mathematical communication between the student and the computer system. This is particularly important since the construction of instructional materials and the manner in which questions are posed to the student often reflect the limitations placed on the student in his construction of answers. Subsequent versions should have an expanded form of response language which is partially controlled by the programmer. Some suggested features are listed here.

1. The programmer should have the capability of defining new functions and making them available to the student. Function names should not be restricted to alphabetic and numeric characters. For example, at selected places it would be convenient to define a function F"(ARG) and allow the student to use F"(x) in his answer. Another function which might be useful to the student is SUM(G[I];I=1,N). Given F(X)=SIN(X), X[0]=1, and X[I+1]-X[I]=H=.2, the student would probably display as much knowledge by constructing the answer (H/2)*(SIN(1) +SIN(2)+2*SUM(SIN(X[I];I=1,4))) as he would in constructing (H/2)*(SIN(1)+2*(SIN(1.2)+SIN(1.4)+SIN(1.6)+SIN(1.8))+SIN(2)). A greater freedom in constructing responses may inspire the student to concentrate more on the concepts involved.
2. The student should have the capability of defining his own functions. Given the greater freedom suggested above in constructing responses, the student would no doubt begin to use

unanticipated function names. As an example, suppose F(x)=(1-1/x) and the author failed to internally define the function F'(x)=1/x↑2. If the student constructs the answer X[K]-F(X[K])/F'(X[K]), the system should interact with the student by asking for a new answer or a definition of F'(ARG).

3. In order to offset some of the difficulties in determining equivalence of expressions, the programmer should have the capability of enabling or disabling standard functions. For example, if the programmer wants to disallow the use of the ARCSIN function, he would turn on a disable flag. At a later point, he may wish to enable ARCSIN.
4. The student should be able to escape the constraints of any mode by entering a computation mode where he could construct and execute programs.

The incorporation of these and similar features requires careful study and planning since a more sophisticated process of matching expressions may be required.

The strategy for processing an individual constructed response (see Figure 2) can be made more effective through a careful study of the student records from the initial experiment. Where little knowledge was initially available on how students respond, it is now possible to begin to enlarge the list of anticipated incorrect responses. Items which are particularly difficult may be changed to allow more than one call for HELP. Unnecessary items may be deleted from the instructional sequence.

With some feel for the difficulty of the instruction in the tutorial modes, it is now possible to begin the construction of multilevel sequences. Existing technology, however, does not guarantee an effective method for choosing or altering the level of instruction for any given student. Examinations can be designed which will allow a student to bypass a section of the instruction or test a student upon completion of a lesson.

Somewhat more definite changes are prescribed for some of the problem modes in order to require a deeper involvement on the part of the student. The current strategy in a problem mode is to ask the student for the equations and parameters in the order they are needed to define the computational procedure. This ordered call for equations tends to serve as an overall prompt or hint, contrary to the philosophy of this mode. A new approach would require the student to work from a basic set of symbols and define the computational procedure in his own way. Problem 3 of the problem mode in Lesson 21 is chosen here to illustrate these concepts. The student must apply Taylor algorithms of orders 1, 2, and 3 to approximate y(2) given $y'=-xy+1/y^2, y(1)=1$. One possible student formulation is presented in the following dialogue.

```
 PROBLEM 3.  (CF. CONTE, EX. 6.3-1)
 LET Y'=F(X,Y)=-X*Y+1/(Y↑2), [A,B]=[1,2], AND Y(1)=1.
 SPECIFY THE PARTIAL DERIVATIVES IN TERMS OF X AND Y.
 FX=F'X=
#-Y
 FY=F'Y=
#-X-2/Y**3
 FXX=F"XX=
#0
 FYY=F"YY=
#6/Y**4
 FXY=F"XY=
#-1
 SPECIFY THE DESIRED ORDER OF THE TAYLOR ALGORITHM (1,2,3,NONE).
 ORDER K=
#3
```

```
 THE TAYLOR ALGORITHM IS Y[I+1]=Y[I]+H*T(X[I],Y[I]).
 YOU HAVE CHOSEN ORDER 3.
 DEFINE T(X,Y) IN TERMS OF H,X, AND Y.  IF YOU WISH, YOU MAY
 USE THE SYMBOLS F, FX, FY, FXX, FYY, AND FXY.
 T(X,Y)=
#F+H*((-Y+FY*F)/2+H*(-2*F+FYY*F**2-Y*FY+F*FY**2)/6)
 SPECIFY N, THE NUMBER OF INTEGRATION STEPS FROM A TO B.  H WILL BE
 COMPUTED AS H=(B-A)/N.  CHOOSE N<101.
 N=
#
```

A typical dialogue using the proposed strategy would be

```
 PROBLEM 3.  Y'=F(X,Y)=-X*Y+1/(Y↑2), [A,B]=[1,2], AND Y(1)=1.
 SPECIFY THE DESIRED ORDER OF THE TAYLOR ALGORITHM.
 ORDER K=
#3
 FORMULATE THE COMPUTATIONAL PROCEDURE Y[I+1]=Y[I]+H*T(X[I],Y[I])
 BY DEFINING AN APPROPRIATE SEQUENCE OF FUNCTIONS (FX,FY,
 FXX,FXY,FYY,T).
 WHICH FUNCTION DO YOU WISH TO DEFINE?
#FY
 DEFINE FY(X,Y)=
#-X-2/Y**3
 WHICH FUNCTION DO YOU WISH TO DEFINE?
#FYY
 DEFINE FYY(X,Y)=
#6/Y**4
 WHICH FUNCTION DO YOU WISH TO DEFINE?
#T
 DEFINE T(X,Y)=
#F+H*((-Y+FY*F)/2+H*(-2*F+FYY*F**2-Y*FY+F*FY**2)/6)
 FORMULATION IS CORRECT.
 SPECIFY N, THE ---
 (etc.)
```

Using the proposed strategy, the student can determine his own path to a correct formulation of a problem. In the above example, one student may choose to define T(X,Y) completely in terms of X and Y and avoid defining the partial derivatives. Another student may wish to define all partial derivatives prior to defining T. If, at any stage, the student types an expression which uses a function not previously formulated, the expression would not be accepted. Each formula entered by the student can be checked in exactly the same way as it is done in the existing system.

No strategy changes are proposed for the investigation modes. In programming these modes, the major difficulty arises in trying to anticipate the needs of the student. From the author's point of view, the investigation modes satisfy the purposes for which they are constructed. Their actual usefulness, in an instructional environment, is yet to be determined. This will be pointed out again in Chapter IV.

Finally, the examples of computer supplied problems demonstrate that this concept can be used in many places in a CAI course in computational mathematics. Technically feasible, their overall usefulness remains to be explored.

CHAPTER IV

EXPERIMENTAL RESULTS AND GENERAL CONCLUSIONS

The Purpose and General History of the Experiment

In keeping with the general objectives of this investigation, this experiment using CAI for computational mathematics was concerned with three basic questions:

1. How do students react to the use of CAI for computational mathematics?
2. What expenditure in time and dollars is required by the teaching methods described in Chapter III?
3. How effective are these methods in teaching computational mathematics?

Although complete answers to these questions would be desirable, the purpose of this experiment was to examine initial trends and indications.

A forty-five item questionnaire was designed to provide some answers to the first question. This questionnaire is presented in Appendix C. In particular, the items on the questionnaire were grouped into three general categories:

1. an evaluation of the structure of the instructional program and the overall and relative merits of the tutorial mode, problem mode and investigation mode,
2. an evaluation of the teletype terminal, and
3. reactions or opinions to miscellaneous items of interest to

the author.

In order to obtain an estimate of the expenditure of resources, records were maintained on the developmental time requirements of the author-programmer, the terminal time requirements of the students, and the computer central processor and peripheral processor time requirements. For this application, the central processor time consists primarily of the execution time required by the PICLS interpreter when it resides as a program in the central memory of the CDC 6500 computer. The peripheral processor time consists primarily of the time required by auxiliary processors to service the terminal and to transfer PICLS course materials from disk storage to central memory for processing by the PICLS interpreter.

Estimates on the effectiveness of CAI are provided by a descriptive comparison of the scores on examinations administered to both the CAI and conventional students. The examinations in Appendix C were designed to test the student on

1. his knowledge of selected theoretical concepts,
2. his ability to use theory in an analysis of problems,
3. his ability to apply algorithms, and
4. his ability to interpret numerical results.

Initial trends and indications, provided by the experiment, will be presented in detail in later sections of this chapter.

Six students were randomly selected from a CS 414 class for the Fall, 1969, CAI experiment. From an operational point of view, the experiment was not without difficulties. Hardward problems on the CDC 6500 combined with software problems of interfacing PICLS with the interactive features of the MACE operating system required an initial curtailment of the available terminal hours. Extra work on weekends and at odd hours was necessary to compensate for system breakdowns and to keep pace with the conventional class. Three of the CAI students volunteered to continue in a program of this type and the other three were returned to the conventional group. As the stability of the hardware-software complex gradually improved, a graduate student volunteer was added to the CAI group. For convenience of discussion and purposes of analysis, the original three CAI students will be referred to by the numbers 34, 35, and 36 and collectively as CAI-1={34,35,36}. The graduate volunteer will be referred to as student 37 and the entire collection of CAI students will be referred to as CAI-2={34,35,36,37}.

For CAI-1, the duration of the experiment was approximately eleven weeks for the completion of twenty-five lessons. This coincided with twenty-nine fifty minute conventional lectures, three examinations, and two holiday periods. For student 37, the duration of the experiment was the amount of time required to cover computer lessons 7-23 after the first examination.

Students 36 and 37 were regularly scheduled for three two-hour sessions each week while students 34 and 35 were scheduled for two three-hour sessions each week. Makeup hours were available upon request during evenings and on weekends.

The CAI students did not attend the conventional lectures but they were required to take the examinations with the conventional class. Upon completion of the experiment, the students filled out a questionnaire and returned to the conventional classroom for the duration of the semester.

Characteristics of the CAI and Conventional Groups

Many variables may be involved in accurately predicting student performance and it is not clear which play a dominant role or which are applicable in predicting the performance of CAI students. The author felt that two available measures might be used to predict achievement in a computational mathematics course:

1. the previous number of semester hours in mathematics which might measure the student's maturity in mathematics, and
2. the cumulative gradepoint in previous mathematics courses which might measure a host of variables such as IQ, aptitude, motivation, etc.

The information on previous mathematics hours and gradepoint was gathered from a questionnaire for each of the thirty-seven students who completed the CS 414 course. The average grade point (gp) and mathematics hours (mh) are listed in Table 8 for the following classes of students:

TOTAL={1,2,...,37}=total population
C*={1,2,...,33}=original conventional group
C=C*-{13,14,29}=conventional students who took all examinations
CAI-1={34,35,36}=original CAI students
CAI-2={34,35,36,37}= total CAI students

The gp and mh were rounded to the nearest one-tenth of a point. Comparisons of CAI-1 and CAI-2 were made with C and subsets of C rather than C* since three students in C* failed to take an examination. Rather than counting the score of zero on the missed examination for students 13, 14, and 29, these students were eliminated from consideration.

Table 8 shows that CAI-1 had a comparatively low gp and mh. This was the result of two effects. First, the lower gp and mh students were the ones to volunteer for retention in the experiment. Secondly, the usual drop out of conventional students was concentrated in the low gp and mh range, thereby increasing the average gp and mh of the remaining conventional group C.

Table 8. Mathematics Background for Various Groups

	TOTAL	C*	C	CAI-1	CAI-2
No. Students	37	33	30	3	4
Average gp	4.8	4.9	4.9	4.1	4.4
Average mh	18.5	18.3	19.0	13.7	20.3

The relative rank of each student is given for mh in Table 9 and gp in Table 10. An examination of the mh and gp figures in Tables 8-10 for the individual members and group averages indicates several things:

1. CAI-1 cannot be expected to compare favorably with C.
2. CAI-2 should compare more favorably with C than CAI-1 compares with C.
3. In terms of both gp and mh, student 37 appears comparable with student 10, but not with any other members of the class.
4. The deletion of {13,14,29} from C* to form C increased the mh of the conventional group.
5. Neither CAI-1 nor CAI-2 are totally representative of C.

This last point is further substantiated by investigating the correlation between mh and gp:

$$r_{mh \times gp}(\text{CAI-2})=.81$$

$$r_{mh \times gp}(\text{C})=.12$$

Table 9. Rankings by Math Semester Hours (mh)

Student	1	2	3	4	5	6	7	8	9	10	11	12	13
mh	23	21	14	17	6	17	17	16	13	40	24	18	14
Rank	5-7	10-13	28-31	17-25	37	17-25	17-25	26-27	32-33	1-2	3-4	14-16	28-31

Student	14	15	16	17	18	19	20	21	22	23	24	25	26
mh	10	17	12	22	17	17	18	18	16	23	22	21	17
Rank	36	17-25	34	8-9	17-25	17-25	14-16	14-16	26-27	5-7	8-9	10-13	17-25

Student	27	28	29	30	31	32	33	34	35	36	37
mh	21	17	11	17	21	23	24	13	14	14	40
Rank	10-13	17-25	35	17-25	10-13	5-7	3-4	32-33	28-31	28-31	1-2

Table 10. Rankings by Math Gradepoint (gp)

Student	1	2	3	4	5	6	7	8	9	10	11	12	13
gp	5.3	6.0	4.5	6.0	5.5	4.8	5.1	5.3	5.0	5.5	4.2	4.2	4.4
Rank	9-11	1-2	23-25	1-2	5-7	17-18	14	9-11	15-16	5-7	32-33	32-33	26-30

Student	14	15	16	17	18	19	20	21	22	23	24	25	26
gp	5.0	4.8	3.6	4.5	4.0	5.5	4.4	4.7	5.4	4.7	5.9	4.7	5.2
Rank	15-16	17-18	36-37	23-25	35	5-7	26-30	19-22	8	19-22	3	19-22	12-13

Student	27	28	29	30	31	32	33	34	35	36	37
gp	4.1	5.3	4.4	4.3	4.7	5.6	4.5	4.4	4.4	3.6	5.2
Rank	34	9-11	26-30	31	19-22	4	23-25	26-30	26-30	36-37	12-13

In CAI-2, gp and mh are highly correlated. This is not true in C. In view of the loss of randomness in the sample and a lack of middle to upper gp and mh students in the sample, a close descriptive analysis of the results was conducted rather than a statistical analysis.

An Analysis of Student Performances

Three examinations totalling twenty-five items were administered to both the CAI students and conventional students. These examinations covered the materials in lessons 1-6, 7-15, and 16-23. The score and relative rank of each student on each examination is available for inspection in Tables 21-23 of Appendix C. In addition, the cumulative average over three examinations is given in Table 24 of Appendix C. The individual performances of the members of CAI-2 and the group averages are shown in Table 11. The average for CAI-1 was lower than the average for C as might be expected since CAI-1 had a much lower average mh and gp. The consistently high performance of student 37 explains the increase in CAI-2 over CAI-1. The average score for CAI-2 was still below that of C, but the substantially higher gp for C may account for this difference in score. One very noticeable point was the uniform decline from their course average for the members of CAI-1 on the third examination. Later remarks may help to explain this decline.

The examination of the final average performances would be of interest in determining the actual importance of mh and gp. Working with the data from Tables 9 and 10 and using the individual course averages from Table 24 as the performance (P), the following correlation coefficients were computed: $r_{mh \times P}(C) = .14$; $r_{mh \times P}(CAI\text{-}2) = .99$; $r_{gp \times P}(C) = .64$ and $r_{gp \times P}(CAI\text{-}2) = .81$.

Table 11. Examination Scores for Various Groups

Group	Exam 1	Exam 2	Exam 3	Average
(34)	38	64	42	48
(35)	68	70	34	57
(36)	56	54	47	52
(37)	92	93	87	91
CAI-1	54	63	41	53
CAI-2	64	70	53	62
C	67	67	67	67

The indication here is that gp was an important predictor of performance in both C and CAI-2 while mh did not appear important in C. The mh effect on the performance of CAI-2 registered astoundingly high. However, the relative importance of gp and mh in CAI-2 is concealed by $r_{gp \times mh}(CAI\text{-}2)=.81$ as reported in the previous section.

In order to look more closely at the effects of mh and gp on the final performance, the linear regression equation

$$\hat{P}=.1765mh+14.42gp-7.181$$

was computed from the mh, gp, and performance data P for the conventional students C. The standard deviation from regression is 11.8 and the correlation between P and $\hat{P}$ is .64. The mh and gp for the various CAI groups in Table 11 were extracted from Tables 8, 9, and 10 in order to predict the expected performance $\hat{P}$ of the CAI students if they had attended the conventional class. These results appear in Table 12. Student 34 performed well below his predicted value, but within one standard deviation. Student 35 performed about as predicted. Student 36 performed better than his predicted value, but within one standard deviation. Student 37 performed more than one standard deviation above his predicted value. Taken collectively, CAI-1 and CAI-2 performed approximately as predicted by the regression equation for C.

Considering each of the twenty-five items on the examinations in Appendix C, CAI-1 scored better than C on seven items and CAI-2 scored better than C on twelve items with one tie. The relative difference between each of the item scores for CAI-2 and C was computed by dividing the absolute difference by the total possible points. This relative difference exceeded .2 for eight items with six in favor of C and two in favor of CAI-2. These items and corresponding scores for CAI-1, CAI-2, and C are given in Table 13. The table indicates that the CAI-1 group had difficulty with some of the theory and a high score by student 37 was not enough to keep the relative difference less than .2. This is not unexpected, considering the lower gp and mh of CAI-1.

Table 12. Predicted and Actual Performance of the CAI Students

Group	Ave mh	Ave gp	$\hat{P}$	P	$P-\hat{P}$
(34)	13	4.4	58.56	48.00	-10.56
(35)	14	4.4	58.73	57.33	- 1.40
(36)	14	3.6	47.20	52.33	+ 5.13
(37)	40	5.2	74.86	90.67	+15.81
CAI-1	13.7	4.1	54.36	52.56	- 1.80
CAI-2	20.5	4.4	59.89	62.08	+ 2.19

Table 13. Exam Items with Large Group Differences

Exam	Item	CAI-1 Ave	CAI-2 Ave	C Ave	Total Possible	Major Purpose of Problem
1	2b	3.3	4.8	7.9	10	Apply Theory
1	3	0	5	11.1	20	Apply Theory
1	4a	8.3	8.8	5.2	10	Interpret Results
2	1a	9	10.5	13.9	15	Understand Theory
2	4b	10	10	7.9	10	Apply Algorithm
3	1a	0	1.3	3.7	5	Understand Theory
3	1b	0	3.8	9.7	15	Understand Theory
3	2b	1.7	3.3	6.5	10	Analyze Problem

Since neither CAI-1 nor CAI-2 appeared to be a good representation of C, it seemed likely that comparisons with subsets of C would yield more information. For each CAI student, subsets of C were formed to collect those students who had similar mh and/or gp characteristics. An average of each subset was then computed to form the average individual C-representatives for each CAI student. Depending on the mh or gp tolerance allowed, each student could have numerous C-representatives. Selected C-representatives for a given mh and gp tolerance were then averaged over the CAI students to form the average group C-representative to be compared with the CAI-1 group and the CAI-2 group.

Denote $gp_k(N)$ as the subset of all students in C which differ from N in CAI-2 by at most .1k gradepoints. For example, from Table 10, it follows that $gp_1(35)=\{3,17,20,30,33\}$ is the set of all students in C who differ from $\{35\}$ by at most .1 gradepoints. In a similar manner, denote $mh_k(N)$ as the set of all students in C who differ from N in CAI-2 by at most k mathematics hours. Thus, from Table 9, $mh_3(35)=\{3,4,6,7,8,9,15,16,18,19,22,26,28,30\}$ is the set of all students in C who differ from $\{35\}$ by at most three mathematics hours. Intersections of subsets were formed to control both mathematics hours and gradepoint, e.g. $gp_1(35)\cap mh_3(35)=\{3,30\}$.

For each N in $CAI\text{-}2=\{34,35,36,37\}$, the following twelve subsets of C were formed: $gp_k(N)$ for $k=0,1,2,3$, $mh_k(N)$ for $k=0,1,2,3$, $gp_k(N)\cap mh_3(N)$ for $k=1,2,3$, and $gp_3(N)\cap mh_2(N)$. Since C is finite, twenty-five of the forty-eight total subsets consist of only one student from C while both $gp_1(37)\cap mh_3(37)$ and $gp_2(37)\cap mh_3(37)$ are empty. For the forty-six non-empty sets, the average gp and mh were computed to form the characteristics of the C-representatives for each CAI student. The examination scores were also averaged to form the performance data of the C-representatives. In this manner, each CAI-1 student has twelve individual C-representatives characterized by gp_0, gp_1, gp_2, gp_3, mh_0, mh_1, mh_2, mh_3, $gp_1\cap mh_3$, $gp_2\cap mh_3$, $gp_3\cap mh_3$, and $gp_3\cap mh_2$. For example, $gp_1(35)$ is a C-representative of student 35 with average $gp=4.4$, average $mh=19$, average exam 1 score=59, average exam 2 score=55, average exam 3 score= 57, and course average=$(59+55+57)/3=57$. Student 37 has only ten individual C-representatives because of two empty intersections. To form the group C-representatives of CAI-1 and CAI-2, the individual C-representative information was averaged over N=34, 35, 36 and N=34, 35, 36, 37, respectively. The results appear in Tables 14 and 15. For example, the mh average of gp_3(CAI-1) in Table 14 was computed as 17.4 by summing the mh averages of $gp_3(34)$, $gp_3(35)$, and $gp_3(36)$ and dividing the total by three. Similar computations were performed for the gp and examination scores.

Table 14 shows twelve group C-representations to be compared with CAI-1. For each representative in Table 14, the number of students is actually the number of contributing students from the corresponding C-representatives for the members of CAI-1. Duplicate students were not counted. For example, there were five students contributing to mh_2(CAI-1) since $mh_2(34)=\{3,9,16\}$ and $mh_2(35)=mh_2(36)=\{8,22,3,9,16\}$. Six conventional students 1, 2, 5, 10, 24, and 32, did not affect the figures in Table 14. Similarly, Table 15 shows the data for the ten group C-representations formed by averaging the respective individual C-representative data for students 34, 35, 36, and 37. Three conventional students 2, 24, and 32, did not affect the figures in Table 15. As previously mentioned, students 13, 14, and 29 were excluded from both tables.

An inspection of Table 14 reveals several trends. First, the previously mentioned uniform decline of CAI-1 on examination three was paralleled only by a decline from the course average in the mh_0, mh_1, and mh_2 representatives. The other representatives showed no large decline. The indication is that students with a weak background in mathematics scored below their course average on the theoretical materials covering numerical differentiation and integration and differential equations. Controlling only the gp, there is a maximum difference in P of seven points between CAI-1 and the corresponding C-representatives, the edge going to the conventional students. The C-representatives also record a stronger mh background, the difference ranging from 2.3 to 3.7. The maximum difference occurs at the gp_3 level where the C-representative has 3.7 more math hours and a slightly higher gradepoint. The major effect appears to be the gradepoint which is in keeping with the value

Table 14. Comparison of CAI-1 with Approximate Representatives in C.

Group	No. of Students	mh Ave	gp Ave	Exam 1 Ave	Exam 2 Ave	Exam 3 Ave	Course Ave (P)	Predicted $\hat{P}$
CAI-1	3	13.7	4.1	54	63	41	53	54
gp_0	2	16.0	4.1	56	53	60	56	55
gp_1	6	16.7	4.2	55	61	54	57	56
gp_2	8	17.0	4.1	55	61	52	56	55
gp_3	12	17.4	4.2	59	67	55	60	56
mh_0	2	13.7	4.7	57	65	50	57	63
mh_1	3	13.3	4.6	62	66	49	59	62
mh_2	5	13.8	4.6	63	72	55	63	62
mh_3	14	15.4	4.9	64	68	63	65	66
$gp_1 \cap mh_3$	3	13.8	4.2	49	68	54	57	56
$gp_2 \cap mh_3$	3	13.8	4.2	49	68	54	57	56
$gp_3 \cap mh_2$	2	13.3	4.2	45	70	52	56	56
$gp_3 \cap mh_3$	3	13.8	4.2	49	68	54	57	56

Table 15. Comparison of CAI-2 with Approximate Representatives in C.

Group	No. of Students	mh Ave	gp Ave	Exam 1 Ave	Exam 2 Ave	Exam 3 Ave	Course Ave (P)	Predicted $\hat{P}$
CAI-2	4	20.2	4.4	64	70	53	62	60
gp_0	3	16.3	4.4	58	56	66	60	60
gp_1	11	17.0	4.4	58	64	61	61	60
gp_2	15	17.0	4.4	59	64	57	60	60
gp_3	23	17.6	4.5	62	69	60	64	61
mh_0	3	20.3	4.9	65	69	62	65	67
mh_1	4	20.0	4.8	69	69	61	66	66
mh_2	6	20.4	4.9	70	74	66	70	67
mh_3	15	21.6	5.0	71	71	71	71	69
$gp_1 \cap mh_3$	----	----	---	--	--	--	--	--
$gp_2 \cap mh_3$	----	----	---	--	--	--	--	--
$gp_3 \cap mh_2$	3	20.0	4.5	56	72	64	64	61
$gp_3 \cap mh_3$	4	20.4	4.5	59	71	65	65	61
C	30	19.0	4.9	67	67	67	67	67

$\hat{P}$ from the regression equation and previously reported correlations. Controlling only the mh, the gp of the C-representation rises to 4.9 as compared to 4.1 for CAI-1. The result is a sizable difference in scores as expected. Controlling both the gp and mh, the difference does not exceed four points.

Table 15 reveals some of these same trends. By controlling only the gp, the difference in scores never exceeds two points. By controlling only the mh, larger differences are detected, but this attributed to a significant increase in the gp for the C-representatives. By controlling both the mh and gp, the difference does not exceed three points with the edge going to the conventional students.

Inspection of Tables 9 and 10 shows that students 10 and 37 were the only two who ranked exceptionally high in both mathematics hours and gradepoint. Table 16 shows the comparison of CAI student 37 with student 10 and also with the class of all conventional graduate students G={8,10}. No striking differences appear in the performances, and the indication is that graduate students performed very well by either method of instruction.

Table 16. Performance at the Graduate Level

Student	mh Ave	gp Ave	Exam 1 Ave	Exam 2 Ave	Exam 3 Ave	Course Ave (P)	Predicted $\hat{P}$
8	16	5.3	89	92	86	89	72
10	40	5.5	89	80	98	89	79
{8,10}	28	5.4	89	86	92	89	76
37	40	5.2	92	93	87	91	75

Although the danger in dealing with small groups of students is realized, the results of the initial experiment indicate that CAI students and conventional students with equal mh and gp performed equally well on examinations.

Observations of the Proctor

During the course of the experiment, the author conducted casual discussions with each CAI student. Some items of interest are noted in this section and may be pertinent in explaining the performance of the CAI students. Judgements concerning the overall motivation of a student were based on observed fluctuations in enthusiasm (especially during periods of excessive hardware failures) and in persistence in learning the concepts (especially during lessons involving difficult subject matter).

Student 34 had noticeable difficulty with the level of the course material. This was further complicated by continual machine failures and the student tended to hurry through the lessons. The student realized his difficulties and, at times, repeated sections of the tutorial mode in order to gain a better grasp of the concepts. This student worked about half of the problems in the problem modes and then hurried to the next lesson. The investigation modes were seldom used. The student had extra-curricular activities which interfered with all of his studies. In particular, he stated that he did not have time to study at all for the first examination. His motivation seemed to be average and remained constant throughout the experiment.

Student 35 found the course material to be challenging, but experienced some serious difficulties in the last eight lessons. He methodically went through the tutorial modes, but he easily gave up when the questions seemed difficult. He learned to put in successive garbage answers when the material was very difficult in order to extract the correct answer from the system. He worked all problems in the problem modes but almost never tried the investigation modes. His overall motivation seemed to be about average but tended to fluctuate with the difficulty of the lessons.

Student 36 found the course to be extremely difficult, but was persistent in his attempts to learn the material. He worked all problems in the problem modes and most suggested problems in the investigation modes. His rate of progress was slow and he often came in during evening hours to do additional work. He realized his weak mathematical background, but attempted to offset this with a high motivation to improve. His motivation, however, fluctuated with the number of hardware failures.

Student 37 had no observable problems. He seemed to work through the tutorial and problem modes with a scientific curiosity. He tried some of his own problems in selected investigation modes. Highly motivated, he found some sections challenging and others easy, but never found the material too difficult.

From general observations and a study of the student records, the following conclusions are tentatively offered:

1. The student operates at a higher efficiency over three two-hour blocks of CAI than over two three-hour blocks.
2. Machine failures are highly disruptive and deter learning.
3. The intrinsic motivation of a CAI student may be the major factor in determining the difference between expected and actual performance.
4. Considering the uniform decline of the CAI-1 students on the last examination, several effects may be present. As

previously mentioned, the C-representatives of CAI-1 based on mh_0, mh_1, and mh_2 also showed a decline. It is possible that a stronger mathematics background is needed to master the course material in lessons 18-23. Since this was the latter portion of the course, it is also possible that an early Hawthorne effect was beginning to disappear. Not to be discounted is the possibility that lessons 18-23 are poorly designed and/or the material is of sufficient theoretical depth to warrant other approaches to teaching the material. The author did experience difficulty in designing instructional sequences for long and involved theoretical developments. Student participation was difficult to envision and, at times, even seemed unnatural.

Additional study is needed to substantiate or repudiate all of these claims.

Results of the Questionnaire

The forty-five item questionnaire displayed in Appendix C was designed to determine the student's reaction to various features of the system. The items on the questionnaire were categorized as follows:

1. Determine the student's reaction to the program structure of the tutorial, problem, and investigation modes--items 2-7, 13, 15, 17-18, 20, 22-24, 27-33, 35-37, and 39.
2. Determine the hardware restrictions of the teletype terminal--items 8, 10-12, 16, 19, and 40.
3. Determine miscellaneous reactions--items 1, 9, 14, 21, 25-26, 34, 38, and 41-45.

The results of the questionnaire were more or less interpreted in terms of ideal conditions. A distribution of responses is given in Table 17. The weights -, 0, and + are to be interpreted in the following manner:

+ means that all students responded favorably to CAI.

0 means that all students took a neutral stand.

(0 or +) means that at least one student responded favorably, at least one took a neutral stand, and no students took a negative stand.

A similar interpretation is placed on - and (- or 0) where negative means an unfavorable response to CAI. Questionable items are those which could not be interpreted because either some students responded favorably while others responded negatively or the item has no + or - interpretation. The individual responses of each student to each item is presented in Table 20 of Appendix C. The discussion here is concerned with responses which have questionable interpretation or are negatively oriented.

The author interpreted the responses to items 30 and 45 as negatively oriented. Three students responded with a 15-30 minute estimate of preparation time for the tutorial mode. Although this may be typical of most students, more time is needed to complete most of the outside reading assignments. Student 34 reported an average of 30-45 minutes for preparation. In general, the students relied heavily on the tutorial mode for an extensive exposure to the course material and did not digest the outside reading prior to the tutorial mode. The students generally agreed that the investigation mode did not provide an outlet for solving their own problems. However, only two students made any serious attempt to use the investigation modes and only one student made extensive use of them. It would appear that a deeper rooted problem exists. Possibly the students did not have extra time or they were not motivated to define their own problems.

Table 17. Distribution of Responses on the Questionnaire

Item Type	-	- or 0	0	0 or +	+	?
Program	30	None	None	3,13,23,31,35	2,4,17,24,27, 29,32,33,36, 39	5,6,7,15,18, 20,22,28,37
Hdwre.	None	None	None	None	8,11	10,12,16,19
Misc.	None	45	None	21,34,42	9,14,25,41	1,26,38,40, 43,44
Totals	1	1	0	8	16	19

The responses to nineteen items had a questionable interpretation. Some of the items were designed to extract information. On others, the students were not in general agreement. In attempting to determine the most useful of the three instructional modes in items 5-7, the opinions were divided. Three students voted to retain the tutorial mode but drop the investigation mode if necessary. However, two of these three students seldom used the investigation mode. Student 36 believed the problem and investigation modes to be the most useful. Of the four students, however, student 36 was the only one to extensively use the investigation mode. There were differing opinions on the difficulty of the linear notation imposed by the teletype terminal and distractions of a noisy typing mechanism. One student felt that he had to concentrate on avoiding syntax errors when typing responses. Two students said that the linear notation made the material more difficult to read and one felt that this difficulty was intensified in the last eight lessons on differentiation, integration, and differential equations. One student seemed bothered by the noise of the typing mechanism.

Specific items regarding course effectiveness resulted in some variation in opinion. Two students felt that half the time they could have gained more from the conventional classroom. It is interesting to note that the performance of both of these students were well above the level predicted by the regression equation in Table 12. Three students agreed that deviations from the textbook made the material more difficult while Student 37 had no difficulty. The tutorial modes seldom clarified the outside reading assignments for Student 36. Student 37 found himself trying to get through the material rather than learning it. This same student said he did not need graphic displays to help him understand the material. Since this student scored high on all examinations, the implication is that he understood the material prior to working through the tutorial mode and that he had very little to gain from CAI. On occasions, he guessed at the answer. The other three students seldom guessed and felt that a graphic display would have helped. Students were divided on a self-evaluation of their own knowledge and their relative performance on examinations.

The Economics of CAI

In terms of author-programmer preparation time, close to one hundred man hours were required to design and implement a lesson and complete the associated tasks. These figures were derived by keeping approximate records of the man hour expenditures for Lessons B and 13-23. Averaging the time over twelve lessons, the following breakdown is reported:

1. initial design (17 hours)
    - specification of lesson objectives
    - specification of subtopics and order of presentation
    - design of examples and exercises
    - specification of format and design of problems for the problem mode and investigation mode
2. coding (24 hours)
3. program checkout (27 hours)
    - data preparation
    - debugging by batch processing
    - debugging by final teletype runs
    - initial revision of the material
4. administration of trial experiment with the lesson (3 hours)
    - proctor the experiment
    - correct errors
5. documentation (3 hours)
    - creation of appropriate pages for the Student Manual
    - creation of the lesson on magnetic tape
6. 20% estimate overhead (19 hours)
    - consultation
    - preparation of questionnaire and examinations
    - correction of errors after the Fall, 1969, experiment
    - unaccounted for activities

Throughout the experiment, PICLS maintained a record of the student terminal time, the central processor time, the peripheral processor time, and the total number of student responses. These figures were accumulated and averaged over the number of participating students and are presented in Table 18. Some records were lost due to machine failure and the average reflects usage for only those students for whom records were available. Thus, the average figures are based only on available information. In those cases where records were lost for all students, the corresponding items are so labelled. If a student failed to use an investigation mode, a zero time was recorded for him. Since the investigation modes were not used by some students, a low average figure appears in most entries of the investigation mode columns. In order to determine the average requirements for a lesson, figures for the three modes were accumulated column-wise and divided by the number of numerical entries in each column. The final averages show that the typical student spent seventy-eight minutes in a tutorial mode, requiring 22.87 seconds of central processor time and 80.94 seconds of peripheral processor time. During this time, the student responded seventy-five times or about once every minute. It should be noted that a response is recorded for each depression of the Return key. Depending on the area of activity, this may or may not imply an actual constructed answer. It does, however, imply that the computer had to service the request from the terminal and that PICLS had to retrieve and process program statements from a disk file.

Based on the current Purdue charges of $275 for each hour of central processing time and $55 for each hour of peripheral processor time, the average computing cost for the typical tutorial mode was (275(22.87)+55(80.94))/3600 or $2.98. Additional calculations appear in Table 19. The prices quoted above are for internal projects. At commercial rates, the costs would be approximately doubled.

Table 18. Average Student and Computer Time Requirements*

Lesson Number	Tutorial Mode	Problem Mode	Investigation Mode	Lesson Totals
A	70/ 18.90/ 93.96/ 93	(does not exist)	(does not exist)	70/ 18.90/ 93.96/ 93
B	47/ 8.15/ 59.48/ 53	14/ 46.69/106.84/ 45	(combined prob-inv)	61/ 54.84/166.32/ 98
1	(records lost)	20/ 14.29/ 41.60/ 35	0/ 0.00/ 0.00/ 0	(records lost)
2	78/ 17.02/ 83.45/ 76	23/ 18.92/ 56.21/ 69	0/ 0.00/ 0.00/ 0	101/ 35.94/139.66/145
3	66/ 23.83/ 81.01/ 67	26/ 20.35/ 67.38/ 76	11/ 2.94/ 12.97/ 13	103/ 47.12/161.36/156
4	(records lost)	46/ 24.95/ 78.78/ 97	37/ 26.75/ 95.41/ 89	(records lost)
5	58/ 17.13/ 40.89/ 52	33/ 33.24/ 77.68/ 80	0/ 0.00/ 0.00/ 0	91/ 50.37/118.57/132
6	73/ 16.09/ 61.28/ 75	46/115.43/186.55/119	0/ 0.00/ 0.00/ 0	119/131.52/247.83/194
7	53/ 18.03/ 67.51/ 78	44/103.34/ 97.94/ 60	15/ 23.74/ 34.26/ 20	112/145.11/199.21/158
8	34/ 43.08/ 97.83/ 55	52/ 57.37/ 96.58/ 56	0/ 0.00/ 0.00/ 0	86/100.45/194.41/111
9	72/ 15.60/ 75.11/ 89	14/ 2.84/ 13.90/ 16	(does not exist)	86/ 18.44/ 89.01/105
10	88/ 22.88/ 66.01/ 70	42/ 49.75/ 56.20/ 53	11/ 20.23/ 33.34/ 18	141/ 92.86/155.55/141
11	41/ 13.73/ 58.16/ 65	27/ 16.19/ 29.11/ 46	25/ 55.50/106.62/ 61	93/ 85.42/193.89/172
12	63/ 25.84/110.23/ 92	41/ 62.12/108.52/ 57	(combined prob-inv)	104/ 87.96/218.75/149
13	76/ 20.63/ 82.10/ 74	20/148.13/301.65/ 28	(combined prob-inv)	96/168.76/383.75/102
14	44/ 14.77/ 50.77/ 56	86/162.06/207.02/225	0/ 0.00/ 0.00/ 0	130/176.83/257.79/281
15	89/ 16.90/ 73.90/ 78	46/ 85.20/123.84/143	4/ 2.88/ 7.81/ 9	139/104.98/205.55/230
16	136/ 18.83/103.33/110	46/ 32.44/ 68.71/ 60	17/ 9.93/ 29.44/ 25	199/ 61.20/201.48/195
17	104/ 19.92/ 95.76/104	42/ 22.30/ 44.95/ 58	4/ 5.76/ 15.15/ 8	150/ 47.98/155.86/170
18	143/ 16.32/ 85.03/ 95	43/ 6.18/ 18.30/ 24	10/ 20.74/ 39.81/ 10	196/ 43.24/143.14/129
19	126/ 18.20/ 85.42/ 90	22/ 8.83/ 22.78/ 18	4/ 19.32/ 38.01/ 9	152/ 46.35/146.21/117
20	62/ 19.95/ 61.01/ 50	36/ 17.47/ 49.45/ 35	6/ 5.06/ 9.88/ 4	104/ 42.48/120.34/ 89
21	117/ 41.75/132.27/ 65	59/353.63/447.66/ 49**	7/ 48.34/ 56.85/ 13	183/443.72/636.79/127**
22	78/ 58.50/107.84/ 69	37/ 86.86/177.45/ 38	0/ 0.00/ 0.00/ 0	115/145.36/285.29/107
23	65/ 39.94/ 89.18/ 58	50/ 98.95/220.82/ 73	(combined prob-inv)	115/138.89/310.00/131
Column Average	78/ 22.87/ 80.94/ 75	38/ 66.15/112.48/ 65	8/ 12.69/ 25.24/ 15	

*Table Format--(Student Minutes)/(CP Seconds)/(PP Seconds)/(No. of Student Responses)

**Excessive CP and PP time due to degraded hardware facilities

Table 19. Computing Costs for CAI

	Cost for Average Use	Adjusted to 60 Minutes	Adjusted to 50 Minutes
Tutorial Mode	2.98	2.29	1.91
Problem Mode	6.77	10.69	8.91
Investigation Mode	1.36	10.20	8.50
All Modes	11.11	5.38	4.48

Table 19 shows that the problem-solving features of the problem and investigation modes come at a high premium. In these modes, arithmetic instructions are abundant and entire blocks of instructions may be executed for a single response in order to provide the student with computational results. In terms of central processing time, an interpretive system heavily penalizes the application of computational mathematics. A direct comparison of the problem and investigation modes with the conventional student's use of the computer for homework assignments was not possible since the conventional students were not required to program all of the numerical methods. Assuming equal effectiveness of the problem-solving facilities of the CAI system and the conventional method of programming, one must eventually determine if the elimination of student programming and debugging in CAI systems is worth the difference in cost.

The basic figure for comparison is $2.98 since the tutorial mode is the portion of the CAI program which was designed as a parallel to the conventional classroom. It should be emphasized that this dollar figure will vary among installations depending on the computer charges needed to run a nonprofit shop. It does, however, appear that an interactive CAI system on a computer which is saturated with background jobs yields a cost which is not totally unreasonable. The \$2.98 for seventy-eight minutes of student terminal time is adjusted in Table 19 to fifty and sixty minutes to provide a better feel for its magnitude. These figures, however, do not include the charges for the terminal and telephone lines. Using \$66.00/month rental for the terminal, \$2.00/month for the line charges, and estimating two hundred usable terminal hours each month, the hourly cost is \$.34 and the total hardware cost becomes \$2.63/hour. This figure does not include course developmental costs and proctoring costs.

Using the student time and hourly cost, certain ratios were computed. In the following computations, only twenty-four CAI lessons were assumed. Lesson A was eliminated from consideration since it teaches the use of the CAI system and the material is not included in the conventional classroom. Its overall effect diminishes as the number of CAI lessons increase. Denote

T_{cai}= student terminal time required for twenty-four tutorial modes=(78)(24)=1872 minutes=31.2 hours

T_c = student classroom time used to cover the equivalent material (twenty-nine fifty-minute lectures)=(50)(29)=1450 minutes =24.17 hours.

The ratio of student time is $R_1=T_{cai}/T_c=1.29$ which means that CAI required about 30% more student time. Based on student time and equal performance $P=P_c=P_{cai}$, the time effectiveness ratio is given by

$$E_1 =(P_{cai}/T_{cai})/(P_c/T_c)=1/R_1=.77$$

which means that CAI was about three-fourths as efficient as the conventional method. Denote

O_{cai}= other costs/hour attributed to developing the CAI course and proctoring students

C_{cai}= total hourly costs for CAI

= $2.63+O_{cai}$

C_c = cost of teaching one conventional student for one hour

The Purdue figure for C_c was not available but Kopstein and Seidel [23] estimate the 1970-71 national average to be \$1.40 for higher education. This estimate is based on cost data prior to 1965 and on a steady annual increment of about 10%. The figure for C_{cai} cannot be computed since O_{cai} is not known, but the hardware cost alone will exceed the allowable break even point. The ratio of total instruction cost was $R_2=R_1(2.63+O_{cai})/1.40=2.42+(.92)O_{cai}$. Based on equal performance, the cost effectiveness ratio is $E_2=(1/R_2)<.42$. The cost of CAI was more than 2.4 times the cost of the conventional method and less than 42% as efficient.

In terms of economics, the conventional method of instruction had a clear cut advantage. However, the total hardware costs can be significantly reduced by designing an instructional system with concentration on efficiency of operation. Central processor time can be significantly reduced by avoiding an interpretive mode of execution. Peripheral processor time can be reduced by avoiding excessive accesses of peripheral storage. In the future, a major effort will be needed to find ways to reduce O_{cai}, particularly the developmental costs.

CHAPTER V

GENERAL FINDINGS AND RECOMMENDATIONS

Specific details have already been presented in the concluding remarks of Chapters II and III and in the various sections of Chapter IV. In this section, an overall summary of the findings is presented along with some recommendations for extending the research. The following points summarize the major findings of this investigation:

1. The feasibility of using CAI for a major portion of the course material has been tentatively established by constructing the program and observing that the average student's terminal behavior on examinations is about the same as representative conventional students. Although the author's manner of presentation might be questioned, the level of difficulty parallels that of the conventional classroom.
2. General difficulty was experienced by the author in designing instruction for the involved theoretical portions of the course dealing with the derivation of numerical methods. In these areas, it was difficult to provide for detailed and meaningful student participation and, at the same time, restrict the instruction to a time period which is reasonably comparable to that of conventional presentation. Successful approaches depend on the ingenuity, experience, and dedication of the instructor. The mathematical maturity of the student seems to

have a significant bearing on the success of the instruction. The student's participation is further hampered by the restricted base of communication which was implemented in the system.

3. The problem-solving aspects, such as exercises, examples, and problems appear natural in this method of instruction.
4. The approximation method described in Chapter II for determining equivalence of expressions was totally successful for this application. It provides the student with a great deal of flexibility in constructing responses within the syntax of the language. The author considers such flexibility to be an important element in the success of CAI in mathematics. It relaxes the restrictions on communication and allows the student to concentrate on concepts. Since it appears externally as an underlying intelligence, the student has confidence in its power to distinguish between correct and incorrect responses. The syntax of the language was limited in this development and recommendations for extensions are detailed in Chapters II and III. A restricted syntax also limits the author's flexibility in designing instructional materials.
5. Although teletype terminals were used in this development, they imposed restrictions on both the author and the student. In some cases, a graphic display is needed to describe the geometry of a numerical method. Even though the students were of a divided opinion on the effects of a linear notation, the author is of the opinion that it is awkward and difficult to

read. Using a natural notation on a CRT display probably would not solve the problems of entering expressions through a keyboard. In particular, the governing rules for forming nested superscripts and subscripts might be complicated.

6. Stability of the hardware-software complex is essential in any production effort. System failures are disappointing to the students. They disrupt the student's concentration and waste his time. In the experiment reported in this paper, it is not known how systems failures may have affected the performance on examinations. Repeated failures in a large scale production effort could have a negative social reaction. Backup systems may be necessary.
7. The design and development of instructional material have some inherent problems. A massive effort in terms of author-programmer time is needed to produce a single-track linear program. This is particularly true in computational mathematics where the definition of variables and assignment of numerical values to variables require a sizable number of supporting arithmetic statements which produce no teletype output. A large number of statements is needed to provide processing support for a single constructed mathematical response. This is true even though an expression may be checked by a single call to the program described in Chapter II. Figure 2 and associated program examples in Chapter III demonstrate this large requirement. Because of these requirements, the overall development failed to accomplish the secondary objectives of

implementing examinations for student evaluation and implementing remedial tracks. Future large-scale developments should be conducted by teams of individuals, representing specialists in instructional design and specialists in subject matter content. Prior to implementation, the project should be reviewed by several institutions in order to gain wide scale acceptance and avoid immediate obsolescence.

8. The problem and investigation modes provide the student with facilities for rapidly solving computational problems. In this respect, the author's approach is considered successful. As pointed out in Chapter III, a revision of the strategy in some of the problem modes may be necessary to provide more challenge to the student. Partial precision and computer-generated problems appear to be useful features in computational mathematics but a careful study has not been conducted. These features place heavier demands on the central processor and the cost of instruction rises.
9. The operational costs for CAI are higher than conventional costs but they are not completely out of range. A carefully designed system could conceivably reduce the computing power costs of the tutorial mode to the cost of the conventional classroom. A major effort is needed to find ways to reduce the developmental costs.
10. A detailed inspection of student scores indicates that CAI students and conventional students with similar mathematics background and mathematics gradepoint will, on the average,

perform equally well.

11. The general student reaction to CAI is positive.

It should be emphasized that conclusions 9-11 are based on a small sample of student histories, examination scores, and the results of a questionnaire. Because the sample was small and because the experiment was plagued with operational problems, the results have to be considered tentative.

The results of the initial experiment opens the way for follow-up experiments of a varying nature. First, several experiments which involve a wide range of students should be conducted to verify the initial results and stabilize the cost estimates. Some experiments should be conducted without the problem and investigation modes. The CAI students would have problem assignments identical to those of the conventional class. In this way, the value of the stand-alone tutorial mode and the effects of the problem mode can be determined. Finally, the tutorial modes should be reconstructed to contain extensive remedial work, examinations, and multiple tracks of instruction. Wherever appropriate, the problem modes should be revised in the manner described at the end of Chapter III. The communication features should be expanded in the manner described in Chapters II and III. All useful experiments conducted up to that point should then be repeated on the extended system.

Of a somewhat different nature, several areas of investigation begin to stem from the current system. The existing course may be supplemented by a graphic display controlled partially by the student and partially by the program. As the student progresses through the material, the program can maintain carefully labelled diagrams or graphs which are pertinent to the discussion. The student may request graphs of his own functions. Hopefully, this would offset some disadvantages of a teletype terminal and lead to deeper understanding of the concepts. Another possibility might be to integrate the current system with the conventional classroom under the control of a instructional management system. Various possibilities can be investigated.

From a broader point of view, the results of research in other areas are needed to create a sophisticated instructional system. A CAI system should have information retrieval capabilities where a student can ask questions and obtain meaningful information. Ideally, the student should be able to communicate in some reasonable subset of a natural language. Character recognition is needed for handwritten communication and speech synthesis for verbal communication. In mathematical systems, the various algorithms of formula manipulation such as symbolic differentiation and integration can be usefully employed. Some standard procedures are desperately needed for distinguishing between conceptual errors and algebraic errors. If these features are combined with advances in learning theory and teaching techniques, we will have some basic tools for building an instructional system.

LIST OF REFERENCES

LIST OF REFERENCES

1. Avner, R. A. and Tenczor, P., The Tutor Manual, CERL Report X-4, Computer-Based Education Research Laboratory, University of Illinois, January, 1969.

2. Balough, R. L., Computer-Assisted Instruction, N68-13897, Clearinghouse for Scientific and Technical Information, U. S. Department of Commerce, January, 1968.

3. Bochner, S. and Martin, W. T., Several Complex Variables, Princeton University Press, 1948.

4. Bunderson, C. V. and Gerry, R., Preparing Educational Material for Computer-Assisted Instruction (mimeographed report), Computer-Assisted Instruction Laboratory, University of Texas, February, 1967.

5. Bunderson, C. V., Dunham, J. L. and Jennings, E., The Role of Computer-Assisted Instruction in University Education, Laboratory for Computer-Assisted Instruction, University of Texas, October, 1967.

6. Caviness, B. F., On Canonical Forms and Simplification (doctoral dissertation), Carnegie-Mellon University, May, 1968.

7. *Charp, S. and Wye, R. E., Computer-Assisted Instruction in a Large* School System, Journal of Educational Data Processing, Vol. 6, No. 1, 1968-69, pp. 28-39.

8. Childs, J. W., A Set of Procedures for the Planning of Instruction, Educational Technology, Vol. VIII, No. 16, 1968, pp. 7-14.

9. Conte, S. D., Elementary Numerical Analysis, McGraw-Hill, Inc., New York, 1965.

10. Entelek Corporation, Computer-Assisted Instruction Guide, Entelek Corporation, Newburyport, Massachusetts, 1968.

11. Fein, L., Thoughts on Computer-Based Instruction, Journal of Educational Data Processing, Vol. 4, No. 4, 1967, pp. 248-253.

12. Feingold, S. L. and Frye, C. H., User's Guide to PLANIT, TM-3055/000/01, System Development Corporation, October, 1966.

13. Feldhusen, J. F. and Szabo, M., A Review of Developments in Computer-Assisted Instruction, Educational Technology, Vol. IX, No. 9, 1969, pp. 32-39.

14. Feldhusen, J. F. and Szabo, M., The Advent of the Educational Heart Transplant, Computer-Assisted Instruction: A Brief Review of Research, Contemporary Education, Vol. XL, No. 5, April, 1969, pp. 265-274.

15. Feurzeig, W. and Bobrow, D. C., MENTOR, A Computer Language for Programmed Discourse (mimeographed report), Bolt, Beranek, and Newman, Inc., 1965.

16. Frye, C. H., CAI Languages: Capabilities and Applications, Datamation, September, 1968, pp. 34-37.

17. Gear, C. W., Computer Organization and Programming, McGraw-Hill, Inc., 1969.

18. Gunning, R. C. and Rossi, H., Analytic Functions of Several Complex Variables, Prentice-Hall, Inc., 1965.

19. Hansen, D. N. and Dick, W., Semiannual Progress Report, Report No. 5, Computer-Assisted Instruction Center, Institute of Human Learning, Florida State University, July, 1967.

20. Hickey, A. E. (ed.), Computer-Assisted Instruction: A Survey of the Literature, Entelek Corporation, Newburyport, Massachusetts, 1968.

21. IBM Corporation, The IBM 1500 Instructional System and Coursewriter II, IBM Corporation, Gaithersburg, Maryland.

22. Kindred, J., Computer Augmented Learning, AD 645 121, Defense Documentation Center, U. S. Department of Commerce, November, 1966.

23. Kopstein, F. F. and Seidel, R. J., Computer-Administered Instruction vs. Traditionally Administered Instruction: Economics, Professional Paper 31-67, Human Resources Office, George Washington University, June, 1967.

24. Korfhage, R., Hochgesang, G., Oldehoeft, A., and Mitzel, M., PICLS, (Purdue Instructional and Computational Learning System), CSD TR 28, Purdue University, October, 1968.

25. Lyman, E. R., A Descriptive List of PLATO Programs, Report R-296, Coordinated Science Laboratory, University of Illinois, Revised July, 1967.

26. Mager, R. F., Preparing Instructional Objectives, Fearon Publishers, Palo Alto, California, 1962.

27. Manacher, G. K., A Content-Evaluating Mode of Computer-Aided Instruction, Interactive Systems for Experimental Applied Mathematics, Academic Press, 1968, pp. 286-293.

28. Martin, W. A., Symbolic Mathematical Laboratory (doctoral dissertation), MAC-TR-36, Massachusetts Institute of Technology, January, 1967.

29. Mitzel, H. E., et al., The Development and Presentation of Four College Courses by Computer Teleprocessing, Pennsylvania State University, June 30, 1967.

30. Oettinger, A. and Marks, S., Educational Technology: New Myths and Old Realities, Harvard Educational Review, Vol. 38, No. 4, 1968, pp. 697-717.

31. Oldehoeft, A. E., Analysis of Constructed Mathematical Responses by Numeric Tests for Equivalence, Proceedings of the ACM Conference, August, 1969, pp. 117-124.

32. Purdue University, MACE Operating System (mimeographed document), Computing Center, Purdue University, December, 1969.

33. RCA, Instructional 70: User's Guide to Instructional Language I, Instructional Systems Division, Radio Corporation of America, Palo Alto, California, December, 1967.

34. Snedecor, G. W., Statistical Methods, The Iowa State University Press, 1946.

35. Stolurow, L. M. and Davis, D., Teaching Machines and Computer Based Systems, In Glaser, R. (ed.), Teaching Machines and Programmed Learning, II, Data and Directions, Department of Audio-Visual Instruction of the National Education Association, Washington, D. C., 1965, pp. 162-212.

36. Suppes, P. S., Jerman, M. and Brian D., Computer-Assisted Instruction: Stanford's 1965-66 Arithmetic Program, Academic Press, 1968.

37. Taylor, E. F. (ed.), ELIZA, A Skimmable Report on the ELIZA Conversational Tutoring System, The Educational Research Center, Massachusetts Institute of Technology, March, 1968.

38. University of California, PILOT 1.5 (mimeographed report), Computer Center, University of California, San Francisco, California.

39. Wilson, E. C., The Knowledge Machine, The Record, Teachers College, Columbia University, Vol. 70, No. 2, 1968, pp. 109-119.

APPENDICES

APPENDIX A

STUDENT MANUAL
FOR A
COMPUTER-ASSISTED COURSE
IN
COMPUTATIONAL MATHEMATICS

Arthur E. Oldehoeft
July, 1969
Second Revision, January, 1970

Computer Sciences Department
Purdue University

Introduction

This manual is a study guide for twenty-five computer-assisted lessons in computational mathematics. The recommended procedure is to sequentially study Lessons A, B, 1,...,23.

Each lesson requires the completion of an outside reading assignment and a computer assignment which deals with the same material. The student may systematically complete each lesson by diligently following the study guides in this manual. General recommended practices are presented in the following paragraphs.

Reading Assignment

The assigned reading should be completed prior to the computer assignment and will always be from the textbook Elementary Numerical Analysis by S. D. Conte. Both the reading assignment and the computer assignment require a prerequisite knowledge of differential and integral calculus and a minimal knowledge of the Fortran computer language. The reading assignment will always cite those materials which should be read prior to beginning the computer lesson.

Computer Assignment

A computer lesson is generally divided into three separate modes of instruction which are described below. A student may begin a particular mode by typing a designated "section name". The section names for each mode will always be listed in the computer assignment. By the time the student has completed Lessons A and B, he will be aware of the significance of each mode of instruction. A computer lesson may be terminated at any point by typing $LOGOFF. If the tutorial mode is terminated in this manner, the student may restart the lesson at a later time at approximately the same point by selecting an appropriate section name from the available list given in the Index at the back of this manual. Due to the manner in which the problem and investigation modes are constructed, the student may restart at the beginning of these modes with very little repetition.

Tutorial Mode

This mode is a programmed instruction presentation of the lesson material and covers all concepts needed for the problem and investigation modes. A variety of examples and exercises are presented to give the student a practical exposure to solving problems. The student is expected to complete the tutorial mode prior to beginning the problem and investigation modes.

Problem Mode

This mode of instruction requires the student to work several standard problems using the computational method studied in the tutorial mode. Problems may be solved with a minimum of computational effort on the part of the student and no programming effort. The problems for each lesson will always be stated in the study guide in order to give the student an opportunity to preanalyze the problem and set up the necessary equations prior to beginning the problem mode. The problem mode may be started any time after completion of the tutorial mode.

Investigation Mode

This mode is optional and the student may use it to solve problems of his own choice. Throughout the tutorial and problem modes, the student will hopefully think of variations of exercises and problems or new and unusual problems. Rapid solution is possible since programming is not required. The student may begin the investigation mode at any time after completion of the tutorial mode.

Student Performance

In each lesson, a simple statement of what is expected of the student on a closed book examination should dictate how much time the student spends in the problem and investigation modes.

Lesson A: Keyboard Orientation

Reading Assignment

Read the first three pages of this manual and the current study guide for Lesson A.

The purpose of this lesson is to familiarize the student with the teletype keyboard and with the sign-on procedure for accessing computer-assisted materials. Upon seating yourself at the teletype, the computer will request the following information:

1. student identification number
2. student password
3. command, section name

A unique student identification number and password is assigned to each student by the instructor. The computer will request this information as the official sign-on procedure. If you have not been assigned an identification number and password, contact your instructor. In order to begin a computer lesson, the student must supply a command and section name. The command will always be $LESSON and the section name must be a legitimate entry specified in the Index of this manual.

As an example, suppose the student with identification number 547 and password AMZ wishes to take Lesson A. The following operations are performed:

1. The student seats himself at the teletype and waits for the message TYPE USER NUMBER:
2. The student types 547.
3. The computer types TYPE PASS WORD.

4. The student types AMZ.
5. The computer types TYPE COMMAND.
6. The student types $LESSON,LOL01..
7. The computer initiates Lesson A.

Computer Assignment

To begin the tutorial mode, use the section name LOL01. There is no problem or investigation mode for this lesson.

Student Performance

Upon completion of this lesson, the student should be able to

1. sign on and off without difficulty for all subsequent lessons;
2. type mathematical expressions;
3. correct typing errors; and
4. apply standard techniques to obtain first estimates of zeros of functions.

Lesson B: Computer Numbers and Computational Error

Reading Assignment

1. Read Conte, pp. 4-11.
2. Review the format for Fortran floating point numbers.

Computer Assignment

The following section names are needed:

1. LOL11 for the tutorial mode
2. LOP01 for the combined problem-investigation mode

Statement of Problems in the Problem Mode. For each of the following problems, the CDC 6500 will simulate a 4, 6, 8, or 15 digit computer. To work each problem, the student must specify

1. the desired precision P=4, 6, 8, or 15
2. numerical values for A, B, and C.

The object of the problems is to observe round-off error and loss of significance.

Problem 1. (see Conte, Ex. 1.3-1) The computer will use p-digit precision to evaluate A+B+C, A/C, A-B, A-B-C, (A*B)/C, B/C, and (B/C)*A.

Problem 2. (see Conte, Ex. 1.4-1, Ex. 1.4-2) Two formulas for finding a root of $A*x^2+B*x+C$ are $(-B+sqrt(B^2-4*A*C))/(2*A)$ and $(-2*C)/(B+sqrt(B^2-4*A*C))$. If 4*A*C is "small" compared to B^2, the effect of C can be lost by using the first formula. For various values of A, B, and C and precisions P=4, 6, 8, and 15, investigate the loss of significance in $B^2-4*A*C$ and the results of both formulas.

Suggested Problems for Investigation Mode (Optional). For this lesson, the problem and investigation modes are one and the same.

Suggested values for Problem 1:

A=.4152	$B=.3572*10^{-4}$	$C=.6321*10^{-6}$
A=1000	B=.4	C=.4
A=.4	B=.4	C=1000
A=.9367	B=.9161	C=.9161

Suggested values for Problem 2:

A=.01	B=1000	C=.004
A=1	B=4	C=0
A=.0001	B=1000	C=1

Student Performance

In order to understand numerical results in future lessons, the student should be fully aware of the concept of round-off error, loss of significance, and how an error may propogate through subsequent calculations. The student should be able to construct his own examples.

Lesson 1: Linear Iteration - Methodology

Reading Assignment

1. Read Conte, pp. 19-21 up to and including the statement, but not the proof, of Theorem 2.1.
2. Read Conte, p. 23.
3. Review the concept of a continuous function.

Computer Assignment

Use the following section names to begin the three available modes:

1. L1L01 for the tutorial mode
2. L1P01 for the problem mode
3. L1I01 for the investigation mode

Problem Mode. Automatic computation is supplied for all problems in the problem modes throughout this course. The student is required to supply the mathematical formulation. Time can be saved by analyzing the problems prior to beginning the problem mode.

Problem 1. The function $F(x)=x-\cos(x)$ has a positive zero P. Find an interval (A,B) and an iteration function G(x) so that

1. $A<P<B$
2. $G(P)=P$
3. $G(x)$ and $G'(x)$ are continuous on (A,B)
4. $\text{abs}(G'(x))< 1$ on (A,B)

You must supply A, B, G(x), and G'(x) for the iteration $x_{k+1}=G(x_k)$.

Problem 2. (see Conte, Ex. 2.1-3) Finding the square root of a number A is equivalent to solving the equation $x^2-A=0$ or finding a zero of $F(x)=x^2-A$. One possible iteration function can be constructed by setting $x^2=A$ and dividing both sides by x to obtain $G(x)=A/x$. Investigate the convergence for various values of A. Which conditions of Theorem 2.1 are violated?

Investigation Mode (Optional). You may use linear iteration on any problem of your own choice. You must supply the iteration equation $x_{k+1}=G(x_k)$ and a starting value x_o.

Suggested Problem 1. Find the zero between 1 and 2 of the function $F(x)=.1*x^2-x*ln(x)$.

Suggested Problem 2. Division by a number $c\neq0$ can be regarded as finding the solution of $F(x)=1/x-c$. Define $G(x)=x*(2-cx)$ and investigate the convergence for various values of c.

Student Performance

Upon completion of this lesson, the student should be able to use various techniques to transform the equation $F(x)=0$ to the form $x=G(x)$ so that all properties of Theorem 2.1 (Conte) are satisfied.

Lesson 2: Linear Iteration - Theory

Reading Assignment

1. Read Conte, pp. 21-22 and 24-26.
2. Work Ex. 2.1-4.
3. Review the mean-value theorem (see Conte, p. 15).

Computer Assignment

Use the following section names to begin the three available modes:

1. L2L01 for the tutorial mode
2. L2P01 for the problem mode
3. L2I01 for the investigation mode

Problem Mode. Work both problems. You must supply G(x), G'(x), and x_o.

Problem 1. (see Conte, Ex. 2.1-1) The cubic polynomial x^3+1.9*x^2-1.3*x-2.2 has a zero P near x=1. Determine an iteration function G(x) and an interval (A,B) so that for x_o in (A,B), the iteration $x_{k+1}=G(x_k)$ will converge to P.

Problem 2. (see Conte, Ex. 2.1-5) The function F(x)=.7-x+.3*sin(x) has a positive zero P. Determine an interval (A,B) and iteration function G(x) so that for x_o in (A,B), $x_{k+1}=G(x_k)$ will converge to P.

Investigation Mode (Optional). You may use linear iteration on any problem of your own choice. You must supply the iteration equation $x_{k+1}=G(x_k)$ and a starting value x_o.

Suggested Problem 1. The linear iteration theorem states sufficient, but not necessary, conditions for convergence. Let

$F(x)=x^3-x^2-x-1$ and $G(x)=x-F(x)/x^2$. Investigate convergence for a wide range of x_0. What conditions of the theorem are violated if we choose $(A,B)=(-10^{10},10^{10})$?

Student Performance

See the student performance for Lesson 1. Given an iteration function G(x), the student should be able to prove that the sufficiency conditions of Theorem 2.1 (Conte) are or are not satisfied. The student should know the formal meaning of "linear convergence" in terms of limits.

Lesson 3: An Acceleration Technique

Reading Assignment

1. Read Conte, pp. 27-30.
2. Work Ex. 2.2-3.

Computer Assignment

Use the following section names to begin the three available modes:

1. L3L01 for the tutorial mode
2. L3P01 for the problem mode
3. L3I01 for the investigation mode

Problem Mode. For each of the problems, you will have to specify the following information:

1. Aitken's delta-squared formula
2. a convergent iteration function G(x)
3. an interval (A,B) on which abs(G'(x))< 1
4. a starting value x_o

Problem 1. (see Conte, Ex. 2.2-1) Find the smallest positive zero of F(x)=2*x-tan(x) using linear iteration and Aitken's delta-squared method.

Problem 2. Find the smallest positive zero of F(x)=.7-x+.3*sin(x) using linear iteration and Aitken's delta-squared method.

Investigation Mode (Optional). You may apply linear iteration and Aitken's delta-squared method to any problem of your own choice. You must specify an iteration equation $x_{k+1}=G(x_k)$, an acceleration formula, and a starting value x_o.

Suggested Problem 1. Let $F(x)=x^2-c$ where $c>0$. For the iteration function $G(x)=c/x$, apply Aitken's process to the iteration $x_{k+1}=G(x_k)$. Compare with the results of Problem 2, Lesson 1.

Suggested Problem 2. Let $F(x)=x^2-c$ where $c>0$. Define the iteration function $G(x)=x-F(x)/F'(x)$. First define the acceleration formula to be $x'_k=x_k$ and find the root. This is equivalent to not accelerating at all. Next, use the standard Aitken's acceleration. Compare the number of iterations for the two methods, say for six digit accuracy.

Student Performance

The student should know Aitken's acceleration formula and given any convergent iteration $x_{k+1}=G(x_k)$, the student should be able to apply the acceleration formula.

Lesson 4: Newton's Method and Quadratic Convergence

Reading Assignment

1. Read Conte, pp. 31-35.
2. Review the linear iteration theorem (Conte, Thm. 2.1).
3. Review Taylor's theorem (Conte, Thm. 1.5 p. 13).
4. Upon completion of the computer lesson, work exercises 2.3-5 and 2.3-6.

Computer Assignment

Use the following section names to begin the three available modes:

1. L4L01 for the tutorial mode
2. L4P01 for the problem mode
3. L4I01 for the investigation mode

Problem Mode. In each problem, you must supply the requested iteration function G(x), the interval (A,B), and a starting value x_o.

Problem 1. (see Conte, Ex. 2.3-1) For any two of the following, find the "smallest positive" zero by Newton's method.

a. f(x)=2*x-tan(x)

b. f(x)=4*cos(x)-exp(x)

c. f(x)=2*cos(x)-cosh(x)

You must supply an interval (A,B) which contains the desired zero but no other zero of f(x), Newton's iteration, and a starting value x_0.

Problem 2. (see Conte, Ex. 2.3-6) $f(x)=(1+1/x)^2$ has a double zero at P=-1. Apply Newton's method and observe that the convergence is linear but not quadratic. Determine (A,B) so that $abs(G'(x)) < 1$. Computation is supplied to display the sequences x_k, $E_k=x_k-P$, E_{k+1}/E_k, and E_{k+1}/E_k^2.

Observe that E_{k+1}/E_k approaches $G'(P)=\frac{1}{2}$ while E_{k+1}/E_k^2 approaches ∞.

Problem 3. (see Conte, Ex. 2.3-6) Apply the modified Newton's method $G(x)=x-2*f(x)/f'(x)$ to the function in Problem 2 and observe that the convergence is quadratic. Determine (A,B) so that $abs(G'(x))<1$. Computation is supplied as in Problem 2. Observe that E_{k+1}/E_k approaches zero and E_{k+1}/E_k^2 approaches $g''(P)/2=1$.

Investigation Mode (Optional). You may use Newton's method or any other iteration $x_{k+1}=G(x_k)$ on any problem of your own choice. You must supply $G(x_k)$ and a starting value x_o.

Suggested Problem 1. $f(x)=(1+1/x)^3$ has a triple zero at $P=-1$. Define a modified Newton's iteration by $x_{k+1}=x-m*f(x_k)/f'(x_k)$. Verify computationally that convergence is linear for $m=1$, 2, 4, 5, and 6, and quadratic for $m=3$. Verify divergence for m greater than 6.

Student Performance

The student is expected to know Newton's method and be able to apply it to practical problems. The student should know the meaning of quadratic convergence in terms of limits.

Lesson 5: The Secant Method

Reading Assignment

1. Read Conte, pp. 39-43.
2. Review Newton's method, the meaning of linear convergence (E_{k+1}/E_k approaches $G'(P)$), and the meaning of quadratic convergence (E_{k+1}/E_k^2 approaches $G''(P)/2$).
3. Work Ex. 2.4-2 in Conte after completion of the computer lesson.

Computer Assignment

Use the following section names to begin the three available modes:

1. L5L01 for the tutorial mode
2. L5P01 for the problem mode
3. L5I01 for the investigation mode

Problem Mode. For each problem, the student must supply the required iteration functions, an interval (A,B) which contains the required zero, and an initial approximation x_o (also x_1 for the secant method).

Problem 1. (see Conte, Ex. 2.4- 1) Draw a graph to estimate the zero of f(x)=x-tan(x) between PI/2 and 3*PI/2. Obtain the zero correct to seven digits by (a) Newton's method and (b) the secant method. A very close estimate of the root P is required for convergence.

Problem 2. (see Conte, Ex. 2.4-3) Find the real positive root of $f(x)=\exp(-x^2)-\log(x)$ correct to seven significant digits using the secant method.

Investigation Mode (Optional). The student may solve any problem of his own choice by supplying an iteration equation $x_{k+1}=G(x_{k-1},x_k)$ and starting values x_o and x_1.

Suggested Problem 1. Investigate the convergence of the secant method for $f(x)=(1+1/x)^2$ where $P=-1$ is a double root. Compare the results with those of Problems 2 and 3 of Lesson 4.

Student Performance

The student is expected to know the formula for the secant method and be able to apply it to practical problems. The student should understand the rate of convergence in terms of limits (see Conte, Ex. 2.4-2).

Lesson 6: Simultaneous Equations

Reading Assignment

1. Read Conte, pp. 43, 44 (last paragraph)-49.
2. Review the concept of a partial derivative from the calculus.
3. Review Taylor's formula with remainder for functions of two variables (see Conte, p. 16).

Computer Assignment

Use the following section names to begin the three available modes:

1. L6L01 for the tutorial mode
2. L6P01 for the problem mode
3. L6I01 for the investigation mode

Problem Mode. For each of the following problems, the student must supply the partial derivatives f_x, f_y, g_x, and g_y and the iteration formulas for Newton's method along with a starting estimate (x_o,y_o).

Problem 1. (see Conte, Ex. 2.5-2) The system $f(x,y)=x^2+y^2-1$, $g(x,y)=x*y$ has four solutions. Use various starting values (x_o,y_o) to find them.

Problem 2. (see Conte, Ex. 2.5-3) Use Newton's method to find solutions to the system $f(x,y)=x^2+x*y^3-9$, $g(x,y)=3*x^2*y-y^3-4$ using starting values (1.2,2.5), (-2,2.5), (-1.2,-2.5), and (2,-2.5). Observe which root the method converges to and the number of iterations required for six significant digit accuracy.

Problem 3. (see Conte, Ex. 2.5-4) Find one solution to the system f(x,y)=x-sin(x)*cosh(y),g(x,y)=y-cos(x)*sinh(y) using Newton's method.

3

Investigation Mode (Optional). The student may work any problem of his own choice by supplying the iteration equations $x_{k+1}=G_1(x_k,y_k)$, $y_{k+1}=G_2(x_k,y_k)$ and a starting value (x_o,y_o).

Suggested Problem 1. Find a solution of the system $f(x,y)=x^2*y^2$, $g(x,y)=x^4+y^4-1$ by Newton's method. Is the convergence quadratic? Explain.

Suggested Problem 2. You will have to use the investigation mode for Lesson 14 (section name L14I01) to solve this problem. Newton's method for three equations in three unknowns $f(x,y,z)=0$, $g(x,y,z)=0$, and $h(x,y,z)=0$ arises from the solution of

$$\begin{bmatrix} f_x & f_y & f_z \\ g_x & g_y & g_z \\ h_x & h_y & h_z \end{bmatrix} \begin{bmatrix} x-x_k \\ y-y_k \\ z-z_k \end{bmatrix} = \begin{bmatrix} -f \\ -g \\ -h \end{bmatrix}$$

where f, g, h, and all partials are evaluated at (x_k,y_k,z_k).
Suppose $f(x,y,z)=x^2+y^2+z^2-1$, $h(x,y,z)=x^2-y^2+z^2$, and $g(x,y,z)=x*y*z$

a. Show that Newton's equations are

$$x_{k+1}=x_k-x_k*(y_k^2+z_k^2-x_k^2*y_k^2)/(2*y_k^2*(x_k^2-z_k^2))$$
$$y_{k+1}=y_k-(x_k^2-z_k^2+y_k^2*z_k^2-x_k^2*y_k^2)/(2*y_k*(x_k^2-z_k^2))$$
$$z_{k+1}=z_k-z_k*(y_k^2*z_k^2-x_k^2-y_k^2)/(2*y_k^2*(x_k^2-z_k^2))$$

b. Use $(x_o,y_o,z_o)=(.2,.8,.8)$ to find the solution.

Student Performance

The student is expected to learn the iteration formulas for Newton's method applied to two simultaneous equations in two variables and be able to apply the method to practical problems.

Lesson 7: Polynomial Equations - Real Roots

Reading Assignment

1. Read Conte, pp. 50-54.

2. Work Ex. 2.6-6 after the computer lesson.

Computer Assignment

Use the following section names to begin the three available modes:

1. L7L01 for the tutorial mode

2. L7P01 for the problem mode

3. L7I01 for the investigation mode

Problem Mode. For each of the following problems, the student must specify the nested multiplication formulas to compute b_n, b_{n-1}, $\ldots, b_o=p(x_k)$, c_n, $c_{n-1}, \ldots, c_1=p'(x_k)$, Newton's iteration in terms of x_k, b_o, and c_1, and a starting value x_o.

Problem 1. (see Conte, Ex. 2.6-1) Use Newton's method for polynomials to find the real root between 0 and -1 of $p(x)=x^3+x+1$.

Problem 2. (see Conte, Ex. 2.6-3) Use Newton's method for polynomials to find a real positive root of

a. $p(x)=x^4+6*x^2-1$

b. $p(x)=3*x^5-2*x^3-2$

c. $p(x)=x^{12}-11*x^{11}+8*x^7-2$.

Investigation Mode (Optional). The student may work any problem of his own choice by specifying for a polynomial, the degree N, the coefficients $a_n, a_{n-1}, \ldots, a_o$, and a starting value x_o.

Suggested Problem 1. Use Newton's method and the sequence of reduced polynomials to determine the multiplicity of the root at x=1 and $x=-1$ of the polynomial $p(x)=x^6+x^5-4*x^4-2*x^3+5*x^2+x-2$.

Student Performance

The student is expected to learn the recursion formulas for Newton's method for polynomials and to be able to apply them to find roots of polynomials.

Lesson 8. Difficulties in Finding Roots of Polynomials

Reading Assignment

1. Read Conte, pp. 55-59.
2. Review Newton's method for polynomials.

Computer Assignment

Use the following section names to begin the three available modes:

1. L8L01 for the tutorial mode
2. L8P01 for the problem mode
3. L8I01 for the investigation mode

Problem Mode. For each problem, you must supply the initial recursion formulas for Newton's method for polynomials and a starting value x_o. After a root is found correct to eight significant digits, use the reduced polynomial to find the next root correct to eight significant digits. When the reduced polynomial is a quadratic, use the quadratic formula to find the remaining two roots. Observe the loss in accuracy caused by error propagating to the reduced polynomials.

Problem 1. (see Conte, Ex. 2.6-4) Four real zeros between -3 and 2 exist for $p(x)=x^4+2.8*x^3-.38*x^2-6.3*x-4.2$. Find these roots, terminating the iteration when $abs(x_{k+1}-x_k)<5*10^{-8}$.

Problem 2. $p(x)=x^4-5*x^2+4$ has exact roots at -2, -1, 1, and 2. Use Newton's method and approximate starting values to find these roots using the sequence of reduced polynomials. Terminate an iteration when $abs(x_{k+1}-x_k)< 5*10^{-8}$.

Investigation Mode (Optional). The specifications are the same as the investigation mode for Lesson 7.

Suggested Problem 1. Conte, Ex. 2.6-5

Student Performance

The student should be aware of possible difficulties when attempting to find the roots of polynomials, e.g. instability, loss of accuracy using the sequence of reduced polynomials, loss of quadratic covergence in case of multiple roots.

Lesson 9: Recursion Formulas for Dividing a Polynomial by a Quadratic Factor and Review of Complex Arithmetic

Reading Assignments

1. Read Conte, pp. 59-60

Computer Assignment

Use the following section names to begin the two available modes:

1. L9L01 for the tutorial mode
2. L9P01 for the problem mode

Problem Mode. $p(x)=x^4-4*x^3+3*x^2+2*x-6$ has two complex roots.

a. Form the quadratic divisor $(x-(1+i))*(x-(1-i))$.

b. Use the recursion formulas to find $b_4, b_3, \ldots, b_o$ and thus determine $Q(x)=b_4*x^2+b_3*x+b_2$ and $R(x)=b_1*(x-S)+b_o$.

c. Observe that $b_1=b_o=0$ which means $R(x)=0$. Hence, $(x-(1+i))*(x-(1-i))$ is an exact divisor of $p(x)$, that is, $1+i$ and $1-i$ are both complex zeros of $p(x)$.

Student Performance

The student should learn the recursion formulas to compute the b_i when dividing a polynomial by a quadratic divisor. The student should observe that if the coefficients of $p(x)$ are real, then complex roots of $p(x)$ must occur in pairs $a+b*i$ and $a-b*i$ and $x^2-2*a*x+a^2+b^2$ is an exact quadratic factor of $p(x)$.

Lesson 10: The Newton-Bairstow Method for Polynomials - Complex Zeros

Reading Assignment

1. Read Conte, pp. 60-64.
2. Review Lesson 9 (recursion formulas for dividing by a quadratic factor).
3. Review Lesson 6, Newton's method for solving simultaneous equations.

Computer Assignment

Use the following section names to begin the three available modes:

1. L10L01 for the tutorial mode
2. L10P01 for the problem mode
3. L10I01 for the investigation mode

Problem Mode. For each of the problems, the student must specify:

a. the recursion formulas to compute each b_i to obtain b_1 and b_o,

b. the recursion formulas to compute each c_i to obtain c_3, c_2, and c_1, and

c. starting values S_o and T_o to define the approximate quadratic factor $x^2-S_o*x-T_o$.

Problem 1. (see Conte, Ex. 2.7-3) Use the Newton-Bairstow method to find a quadratic factor of $p(x)=x^4+3*x^2+1$. An approximate root is z=1.6*i.

Problem 2. (see Conte, Ex. 2.7-3) $p(x)=x^4+2*x^3+6*x^2-13*x+48$ has a complex zero near z=1+sqrt(3)*i. Use the Newton-Bairstow method to find a quadratic factor of p(x).

Problem 3. $p(x)=2*x^3-2.0545802*x^2-.9491684$ has a complex zero near z=.15+.8*i. Use the Newton-Bairstow method to find a quadratic factor of p(x).

Investigation Mode (Optional). This mode provides automatic computation for either Lin's method or the Newton-Bairstow method. The student must specify the method, the degree of the polynomial, the coefficients of the polynomial, and initial estimates S_o and T_o for the quadratic divisor $x^2-S_o*x-T_o$.

Suggested Problem 1. $p(x)=x^4-4*x^3+10*x^2-12*x+9$ has a double complex zero near $z=.9+1.4*i$. Does the Newton-Bairstow method converge quadratically?

Suggested Problem 2. Conte, Exercises 2.7-2 and 2.7-5.

Student Performance

The student should learn the recursion formulas for the Newton-Bairstow method and be able to apply the method to practical problems.

Lesson 11: The Solution of Linear Systems by Elimination

Reading Assignment

1. Read Conte, pp. 156-163.

Computer Assignment

Use the following section names to begin the three available modes:

1. L11L01 for the tutorial mode
2. L11P01 for the problem mode
3. L11I01 for the investigation mode

Problem Mode. In both problems, the computer will maintain six significant digit accuracy throughout the computation. The object of the problems is to observe the advantage in using the method with pivoting. Both problems deal with the linear system Ax=B given by the augmented matrix

$$\begin{bmatrix} .000003 & .213472 & .332147 & .235262 \\ .215512 & .375623 & .476625 & .127653 \\ .173257 & .663257 & .625675 & .285321 \end{bmatrix}$$

Problem 1. Solve the above system by elimination without pivoting by using the sequence of row operations

(Row J)+M*(Row I) replaces (Row J).

You must specify M, I, and J for each row operation.

Problem 2. Solve the above system by elimination with pivoting by using the row operations

Interchange (Row I) and (Row J)

(Row J)+M*(Row I) replaces (Row J)

You must specify the operation to be performed and the corresponding values of I, J, and M.

Investigation Mode (Optional). The student may solve any linear system of his choice using 4, 6, or 8 significant figure accuracy throughout the computation. The student specifies:

1. the precision 4, 6, or 8
2. the dimension of the system Ax=B
3. the elements of A and B

Computation is supplied by the computer as the student directs any of the following sequence of operations:

1. Interchange (Row I) and (Row J)
2. Replace (Row J) by (Row J)+M*(Row I)
3. Begin the back-substitution
4. Print the current augmented matrix
5. Restart the problem with the original A and B
6. Input a new A and B
7. Terminate the investigation mode

Suggested Problem 1. Use elimination to find the solution of the system

$$\begin{bmatrix} 6 & 15 & 9 & 13 \\ 2 & 17 & 11 & 1 \\ 4 & 10 & 14 & 8 \\ 5 & 12.5 & 7.5 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 19 \\ 11 \\ 7 \end{bmatrix}$$

Note what happens after forming zeros in positions A_{21}, A_{31}, and A_{41}. Explain!

Suggested Problem 2. This example will be encountered again in Lessons 12 and 13. Note the variation in the solution by using different precision arithmetic.

$$\begin{bmatrix} 2.53423 & 8.93734 & 4.37526 \\ 1.02435 & 3.61254 & 3.22463 \\ .853217 & 3.00906 & 7.29341 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.24763 \\ 2.55174 \\ 6.15257 \end{bmatrix}$$

Lesson 12: Evaluation of Determinants and Matrix Inversion

Reading Assignment

1. Read Conte, pp. 169-174.
2. Review the method of elimination.
3. Work Ex. 5.5-3 after the completion of the computer lesson.

Computer Assignment

Use the following section names to begin the two available modes:

1. L12L01 for the tutorial mode.
2. L12P01 for the problem-investigation mode.

Problem-Investigation Mode. You may specify any problem of your own choice or you may request the computer to generate a matrix A with random integers as elements. For each problem, you must

1. specify the dimension N=2, 3, or 4 for the matrix A;
2. specify the elements of A (or ask for random elements);
3. use elimination to reduce the N by 2*N augmented matrix A | I to triangular form; and
4. use back-substitution to compute $B=A^{-1}$

Suggested Problem 1. Conte, Exercises 5.5-1, 5.5-2, and 5.5-4.

Suggested Problem 2. Find the inverse of the coefficient matrix in the Suggested Problem 2 of the investigation mode for Lesson 11.

Student Performance

Upon completion of this lesson, the student should be able to apply the method of elimination to find the inverse of a given matrix A, check the accuracy of A^{-1} by comparing AA^{-1} with the identity matrix I, and given a system A*x=B, compute the solution $x=A^{-1}B$.

Lesson 13: Errors and Conditioning

Reading Assignment

1. Read Conte, pp. 163-169.
2. Review the method of elimination.

Computer Assignment

Use the following section names to begin the two available modes:

1. L13L01 for the tutorial mode
2. L13P01 for the problem-investigation mode

Problem-Investigation Mode. You may specify any problem of your own choice or you may request the computer to generate a problem for you. In the latter case, the computer will generate a matrix which is ill-conditioned. For each problem, you must

1. specify the dimension, N=2, 3, or 4, of the matrix A;
2. specify the arithmetic precision, M= 4, 6, or 8 significant digits, for all internal computations; and
3. specify the elements of the matrix A and vector B for the system A*x=B or request the computer to generate them for you.

To solve a problem, you must direct the computer through some sequence of the activities listed below.

1. Interchange rows.
2. Perform the current stage of elimination.
3. Compute the normalized determinant (assuming the matrix has been reduced to triangular form).
4. Compute the solution x after reaching a triangular form.
5. Compute the residual vector after finding x.
6. Find the solution to the error system A*E=R and compute the

improved solution $x_{new}=x+E$ after completion of step 5.

Upon completion of a problem, the student may elect to change the precision M and rework the same problem.

Suggested Problem 1. Conte, Exercises 5.4-1, 5.4-2, 5.4-3 and 5.4-4.

Suggested Problem 2. Rework Suggested Problem 2 of the investigation mode for Lesson 11.

Student Performance

Upon completion of this lesson, the student should be able to use elimination to find norm $|A|$ and determine if the system is ill-conditioned, set up and solve the error system A*E=R, and thus attempt to improve the solution.

Lesson 14: Iterative Methods for Solution of Linear Systems

Reading Assignment

1. Read Conte, pp. 191-195.

Computer Assignment

Use the following section names to begin the three available modes:

1. L14L01 for the tutorial mode
2. L14P01 for the problem mode
3. L14I01 for the investigation mode

Problem Mode. For any two of the following problems, investigate the convergence of both the method of simultaneous displacements and the method of successive displacements. You must specify the iteration equations and your choice of starting values.

Problem 1.

$$\begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1.25 \\ -.5 \\ 2.75 \end{bmatrix}$$

Problem 2.

$$\begin{bmatrix} 1 & .5 & .5 \\ .5 & 1 & .5 \\ .5 & .5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 4.0 \\ 3.5 \end{bmatrix}$$

Problem 3.

$$\begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Investigation Mode (Optional). You may use any iterative method to solve a system of equations of your own choice (linear or nonlinear). You must specify

1. the number of equations $0<N<7$,
2. the N iteration equations in terms of x1,...,xN, and
3. the starting values for each variable.

Suggested Problem 1. Investigate the convergence of both iterative methods for the lower triangular system

$$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 3 & 0 \\ 3 & 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ -3 \end{bmatrix} .$$

How many iterations are required? Can you generalize to an n x n triangular system?

Student Performance

Upon completion of this lesson, the student should know both the method of simultaneous displacements and successive displacements and be able to apply them to a linear system of equations.

Lesson 15: Convergence of Iterative Methods for Linear Systems

Reading Assignment

1. Read Conte, pp. 199-203.
2. Review the methods of simultaneous displacements and successive displacements.

Computer Assignment

Use the following section names to begin the three available modes:

1. L15L01 for the tutorial mode
2. L15P01 for the problem mode
3. L15I01 for the investigation mode

Problem Mode. If either the row or column sum criteria is satisfied, we are assured of convergence of both the method of simultaneous displacements and the method of successive displacements. If neither is satisfied, a method may or may not converge. For each of the problems, investigate convergence of both methods. You must supply the iteration equations and starting values.

Problem 1.

$$\begin{bmatrix} 1 & 1 & 1 \\ .5 & 2 & 2 \\ .25 & .5 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.75 \\ 2.75 \\ 4.25 \end{bmatrix}$$

Problem 2.

$$\begin{bmatrix} 2 & -1 & -.75 \\ 3 & 4 & .75 \\ -3 & .75 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2.075 \\ .225 \\ 4.1 \end{bmatrix}$$

Problem 3.

$$\begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} PI \\ PI/2 \\ PI/4 \end{bmatrix}$$

Investigation Mode (Optional). The specifications are the same as for the investigation mode for Lesson 14.

Suggested Problem 1. Observe the rapid convergence of both methods for the system

$$\begin{bmatrix} 50 & .1 & 7 \\ .02 & 10 & 1.4 \\ 13 & -13 & 31 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 17 \\ 9 \\ 23 \end{bmatrix} .$$

How many iterations are required for six digit accuracy and for eight digit accuracy? Can you form other systems for which convergence is rapid?

Student Performance

The student is expected to know both the method of simultaneous displacements and the method of successive displacements. The student should be able to apply both the row sum and column sum criteria to predict convergence or divergence.

Lesson 16: Numerical Differentiation

Reading Assignment

1. Read Conte, pp. 108-113.
2. Review Taylor's formula with remainder (see Conte, p. 15).
3. Work Ex. 4.1-6 after completion of the computer assignment.

Computer Assignment

Use the following section names to begin the three available modes:

1. L16L01 for the tutorial mode
2. L16P01 for the problem mode
3. L16I01 for the investigation mode

Problem Mode. For each of the problems listed below, find a value of h which will yield the specified accuracy when using the approximations

$$D(h)=(f(x_{i+1})-f(x_{i-1}))/2h = (f(x_i+h)-f(x_i-h))/2h$$

$$D2(h)=(f(x_{i+1})-2f(x_i)+f(x_{i-1}))/h^2 = (f(x_i+h)-2f(x_i)+f(x_i-h))/h^2$$

For example, see Table 4.1, Conte, p. 112.

In this mode, the student enters a value of h and the values of D(h) or D2(h) will be printed. The student must experimentally find a value of h for which the combination of truncation error and round-off error are small enough to yield the specified accuracy.

Problem 1. (see Conte, Ex. 4.1-4) $f(x)=\cosh(x)$, $x_i=1.4$.

Desired accuracy: $| f'(x_i)-D(h) | < .5*10^{-9}$ and $| f''(x_i)-D2(h) | <.1*10^{-6}$

Problem 2. $f(x)=\sin(x)$, $x_i=.4$

Desired accuracy: $| f'(x_i)-D(h) | < .1*10^{-9}$ and $| f''(x_i)-D2(h) | <.3*10^{-7}$

Problem 3. $f(x)=\exp(x)/\sin(x)$, $x_i=1.1$

Desired accuracy: $| f'(x_i)-D(h) | <.5*10^{-9}$ and $| f''(x_i)-D2(h) | < .1*10^{-5}$

Problem 4. $f(x)=sqrt(2*sin^2(x)+cos^2(x)),x_i=5.3$

Desired accuracy: $| f'(x_i)-D(h) | < .5*10^{-9}$ and $| f''(x_i)-D2(h) | < .5*10^{-7}$

Investigation Mode (Optional). You may apply any numerical differentiation formula D(h) to any function f(x). You must supply

1. f(x),
2. D(h) to approximate $f'(x_i)$ or $f''(x_i)$,
3. the first tabular point x_o, and
4. the spacing h of the tabular points and the total number ($N<10$) of tabular points.

Upon completion of step 4, the computer will print the table of tabulated function values (i=0,...,N): i, x_i, and $f(x_i)$. Each time the student defines a value for i, the computer will print D(h). By typing STOP, the student may restart the problem at any one of the four steps.

Suggested Problem 1. The instability of numerical differentiation can be displayed by simple examples where the slope and/or concavity of a function change rapidly. Consider $f(x)=-2x^4+4x^2+16$. Note that f(x) is symmetric about 0 with f(0)=16, $f(\pm 1)=18$, and $f(\pm 2)=0$. In general, it is more difficult to approximate f'(1) than f'(0) since f(x) changes rapidly at x=1. For various values of h, approximate f'(0) and f'(1) by the three formulas:

$D(h)=(f(x_{i+1})-f(x_{i-1}))/2h$ $O(h^2)$-approximation

$D(h)=(f(x_{i+1})-f(x_i))/h$ O(h)-approximation

$D(h)=(-3f(x_i)+4f(x_{i+1})-f(x_{i+2}))/2h$ $O(h^2)$-approximation

For various values of h, approximate f''(0) and f''(1) by the $O(h^2)$-approximation $D(h)=(f(x_{i-1})-2f(x_i)+f(x_{i+1}))/h^2$.

Student Performance

The student is expected to know the D(h) and D2(h) operators used in Problems 1-4 and their respective orders. The student should be aware of the effects of round-off when h is very small and be able to apply the formulas to practical problems.

Lesson 17: Extrapolation to the Limit

Reading Assignment

1. Read Conte, pp. 114-119.
2. Review Taylor's formula with remainder (see Conte, p. 15).
3. After completion of the computer assignment, work Exercises 4.2-1, 4.2-4, and 4.2-6.

Computer Assignment

Use the following section names for the three available modes:

1. L17L01 for the tutorial mode
2. L17P01 for the problem mode
3. L17I01 for the investigation mode

Problem Mode. In each problem, you will be supplied with a set of tabulated values for a function f(x). You must supply the numeric values or expressions to effect extrapolation to the limit in the table.

Problem 1. Use extrapolation to the limit to approximate f'(.4) where f(x)=sinh(x).

i	x_i	f_i
0	.398	sinh(.398)
1	.399	sinh(.399)
2	.400	sinh(.400)
3	.401	sinh(.401)
4	.402	sinh(.402)

Problem 2. (see Conte, Ex. 4.2-3) Use extrapolation to the limit to approximate f'(.5) where f(x)=sin(x)/x.

i	x_i	f_i
0	.3	sin(.3)/.3
1	.4	sin(.4)/.4
2	.45	sin(.45)/.45
3	.5	sin(.5)/.5
4	.55	sin(.55)/.55
5	.6	sin(.6)/.6
6	.7	sin(.7)/.7

Problem 3. Use extrapolation to the limit to approximate f'(0) where f(x)=exp(-x)*sin(x).

i	x_i	f_i
0	-.16	exp(.16)*sin(-.16)
1	-.08	exp(.08)*sin(-.08)
2	-.04	exp(.04)*sin(-.04)
3	-.02	exp(.02)*sin(-.02)
4	-.01	exp(.01)*sin(-.01)
5	0	0
6	.01	exp(-.01)*sin(.01)
7	.02	exp(-.02)*sin(.02)
8	.04	exp(-.04)*sin(.04)
9	.08	exp(-.08)*sin(.08)
10	.16	exp(-.16)*sin(.16)

Investigation Mode (Optional). You may apply extrapolation to the limit to approximate f'(Z) for your own choice of f(x). You must supply

1. the value of Z,
2. the value of h for the initial approximation D(h),
3. the number of entries $(x_o,f_o),\ldots,(x_N,f_N)$ in the table $(N<10)$, and
4. either the function values $f_o,\ldots,f_N$ or the function f(x) from which the computer will compute $f_o,\ldots,f_N$.

Extrapolated values will be computed line by line in a table of the form

$$\begin{array}{lllll}
D(h) & & & & \\
D(h/2) & D1(h/2) & & & \\
D(h/4) & D1(h/4) & D2(h/4) & & \\
\cdot & \cdot & \cdot & \cdot & \\
\cdot & \cdot & \cdot & \quad\cdot & \\
\cdot & \cdot & \cdot & \qquad\cdot & \\
D(h/2^m) & D1(h/2^m) & D2(h/2^m) & \ .\ \ .\ \ . & Dm(h/2^m)
\end{array}$$

Suggested Problem. Use extrapolation to the limit to approximate f'(0) and f'(1) for f(x) given in Suggested Problem 1 of the investigation mode in Lesson 16.

Student Performance

Given a function or table of function values, the student should be able to apply extrapolation to the limit and state the order of any approximation in the table.

Lesson 18: Numerical Integration - The Trapezoidal Rule

Reading Assignment

1. Read Conte, pp. 119-124.
2. Review Rolle's theorem (see Thm. 1.3, Conte, p. 15).
3. Review the second theorem of the mean for integrals (see Conte, p. 15).

Computer Assignment

Use the following section names to begin the three available modes:

1. L18L01 for the tutorial mode
2. L18P01 for the problem mode
3. L18I01 for the investigation mode

Problem Mode. For each problem in this mode, use the trapezoidal rule to approximate the integral of f(x) from A to B. To solve the problem, you must specify

1. the error $E(h)=-h^2*f''(z)/12$ in terms of h and z where $A<z<B$,
2. a value of h analytically determined so that max $|E(h)|<\epsilon$ on [A,B] for a prescribed ϵ,
3. the number of subdivisions N based on your value of h, and
4. the formulas for the trapezoidal rule based on Int(N/4)+1, Int(N/2)+1, and N subdivisions of [A,B] in terms of f_i and h.

Problem 1. f(x)=sqrt(x)+1/sqrt(x), [A,B]=[1,2], and $\epsilon=.5*10^{-2}$

Problem 2. $f(x)=\exp(-x^3)$, [A,B]=[0,1], $\epsilon = .5*10^{-2}$

Investigation Mode (Optional). You may apply the trapezoidal rule to approximate an integral of your own choice. You must specify f(x), A, B, and the number of subdivisions $N < 41$.

Suggested Problem 1. $f(x) = \frac{2}{2+\sin(10*PI*x)}$ is a periodic function with period equal to .1. One danger in using equally spaced points for integration is discovered by the numerical integration of periodic functions. Investigate this effect by using the trapezoidal rule with N=30, 35, and 40 subdivisions (31, 36, and 41 points) to approximate $\int_0^1 f(x)dx$. The exact value is 2/sqrt(3).

Suggested Problem 2. f(x)=abs(x) has a discontinuity in the first derivative at x=0. So the error formula does not apply if the interval for integration contains 0 as an interior point. Yet the method is exact if we subdivide the interval so that 0 is an end point of a subdivision. Investigate this effect by using the trapezoidal rule to approximate $\int_{-3/4}^{1} abs(x)dx$. Use an even and odd number of points. Explain the results.

Student Performance

The student should know the trapezoidal formula and be able to apply it to approximate definite integrals. The student should know the error formula and be able to analytically determine a value of h, for simple functions, so that the absolute error is less than some prescribed tolerance.

Lesson 19: Romberg Integration

Reading Assignment

1. Read Conte, pp. 126-131.
2. Review the trapezoidal rule.
3. Review Taylor's formula with remainder (see Conte, p. 15).

Computer Assignment

Use the following section names to begin the three available modes:

1. L19L01 for the tutorial mode
2. L19P01 for the problem mode
3. L19I01 for the investigation mode

Problem Mode. For the following problems, you are to state the trapezoidal rule for the specified values of N and the formulas for extrapolation to the limit. Numerical values will be supplied as the formulas are constructed.

Problem 1. Use Romberg integration to approximate the integral of f(x)=sin(x)/x from 0 to 1 using N=1, 2, and 4 subdivisions. Note f(0)=1 by investigating the limit.

Problem 2. Use Romberg integration to approximate the integral of f(x)=ln(x) from 1 to 3 using N = 1, 2, 4, and 8 subdivisions.

Investigation Mode (Optional). The student may use Romberg integration to approximate his own choice of the integral of f(x) from A to B. The student supplies f(x), A, B, and the initial number of subdivisions N<20. h=(B-A)/N will be computed and the extrapolation results will be printed line by line for h/2, h/4, etc. using 2*N, 4*N, etc. subdivisions until the number of subdivisions exceeds 40.

Suggested Problem 1. See Suggested Problem 1 in the investigation mode for Lesson 18. Use Romberg integration.

Suggested Problem 2. See Suggested Problem 2 in the investigation mode for Lesson 18. Use Romberg integration.

Student Performance

The student should be able to apply Romberg integration to approximate the value of an integral. This requires defining the formulas needed to construct the Romberg integration table. The student should know the order of the approximation for any entry in the table.

Lesson 20: Numerical Integration - Simpson's Rule

Reading Assignment

1. Read Conte, p. 131 (first half page).
2. Read Conte, pp. 134 - 137, beginning with formula 4.57.
3. Review the trapezoidal rule, Romberg integration, and the second theorem of the mean for integrals (see Conte, p. 15).

Computer Assignment

Use the following section names to begin the three available modes:

1. L20L01 for the tutorial mode
2. L20P01 for the problem mode
3. L20I01 for the investigation mode

Problem Mode. For the problems, approximate the integral of f(x) from A to B using N=3 and N=5 points (2 and 4 subdivisions) to obtain the $O(h^2)$-trapezoidal estimates T0[0] and T0[1]. Then use simple extrapolation to obtain the improved estimate T1[1]. Finally, use Simpson's rule with 5 points to approximate the integral. The results of T1[1] and Simpson's rule should be the same.

Problem 1. (see Conte, Ex. 4.5-1) f(x)=sin(x)/x, f(0)=1,A=0, B=1.

Problem 2. (see Conte, Ex. 4.5-4) $f(x)=\exp(-x^2)*\sin(x)$, [A,B]= [0,1].

Problem 3. (see Conte, Ex. 4.5-2) $f(x)=\exp(-x^3)$, [A,B] = [0,1].

Investigation Mode (Optional). You may use Simpson's rule to approximate the integral of your own choice of f(x) from A to B. You must supply f(x), A, B, and $N < 11$ (number of 2h-length intervals).

Suggested Problem 1. See Suggested Problem 1 in the investigation mode for Lesson 18. Use Simpson's rule.

Suggested Problem 2. See suggested Problem 2 in the investigation mode for Lesson 18. Use Simpson's rule.

Student Performance

The student should be able to state and apply Simpson's rule for 2*N subdivisions to approximate an integral. He should be able to state the error formula E(h) and, for simple functions, choose h so that $\max |E(h)| < \epsilon$ for a specified ϵ.

Lesson 21: Numerical Integration of Ordinary Differential Equations by Taylor Series Approximations

Reading Assignment

1. Read Conte, pp. 212-217.
2. Review Taylor's formula (see Conte, p. 15).
3. Review the definition of the order of an approximation (see Conte, p. 115).

Computer Assignment

Use the following section names to begin the three available modes:

1. L21L01 for the tutorial mode
2. L21P01 for the problem mode
3. L21I01 for the investigation mode

Problem Mode. In the following problems, you are to numerically approximate the true solution y(x) on [A,B] by Taylor's algorithm of orders 1, 2, and 3 for the given y'=f(x,y) and y(A). Experiment with several values of h. As h is chosen smaller, the approximations become more accurate. For each problem, you must determine

1. f_x, f_y, f_{xx}, f_{yy}, and f_{xy},
2. $y_{i+1}=y_i+h*T(x_i,y_i)$ as the Taylor algorithm and specify T(x,y) for orders 1, 2, and 3, and
3. a value N=step size for computation of x_i, y_i for i=0,1,...,N.

Problem 1. (see Conte, Ex. 6.3-2) Let y'=f(x,y)=2y, [A,B]=[0,1], and y(0)=1. The exact solution is y(x)=exp(2x).

Problem 2. (see Conte, Ex. 6.3-1) Let $y'=f(x,y)=-xy+1/y^2$, [A,B]=[1,2], and y(1)=1.

Investigation Mode (Optional). You may use Taylor's algorithm of orders 1, 2, and 3 to solve any problem $y'=f(x,y)$ over the interval $[A,B]$. You must specify

1. $y'=f(x,y)$,
2. the desired order and corresponding expressions for f_x, f_y, f_{xx}, f_{yy}, and f_{xy},
3. initial conditions $(A,y(A))$ and the final value of $x=B$, and
4. h so that $N=(B-A)/h$ is an integer less than 101.

Suggested Problem 1. Consider the initial value problem $y'=y^2$ with $y(1/2)=2$. The exact solution is $y(x) = \frac{1}{1-x}$. Note that exact solution y and all of its derivatives $y'=f$, $y''=f'$, etc. have a singularity at $x=1$. Thus integration over the interval $[\frac{1}{2},1.4]$ to approximate $y(1.4)=-2.5$ violates the assumptions of continuity on y, f, f' etc. Use various values of h and Taylor's algorithm of orders 1, 2, and 3 to see how integration over singularities behaves. Then repeat the integration starting at $y(1.2)=-5$ to avoid the singularity at $x=1$.

Student Performance

Given an initial value problem $y'=f(x,y)$, $y(A)$ specified, the student should be able to apply the computational method for Taylor's algorithm of orders 1, 2, or 3 to approximate $y(x)$ over an interval $[A,B]$.

Lesson 22: Second Order Runge-Kutta Methods

Reading Assignment

1. Read Conte, pp. 220-225.
2. Work Ex. 6.5-3 in Conte.
3. Review Taylor expansions of functions of one and two variables (see Conte, pp. 15-16).

Computer Assignment

Use the following section names to begin the three available modes:

1. L22L01 for the tutorial mode
2. L22P01 for the problem mode
3. L22I01 for the investigation mode

Problem Mode. For each problem, find the solution to the initial value problem over the specified interval. Use A=0, B=1, C=D=1/2 for the modified Euler's method and A=B=1/2, C=D=1 for the improved Euler's method. The general second order Runge-Kutta method is

$$y_{i+1}=y_i+A*K1+B*K2$$
$$K1=h*f(x_i,y_i)$$
$$K2=h*f(x_i+C*h,y_i+D*K1)$$

The student must specify $K1(x_i,y_i)$, $K2(x_i,y_i)$, the formula for y_{i+1}, and any value of $h \geq .01$ so that the number of integration steps N is an integer.

Problem 1. (see Conte, Ex. 6.5-2) Let $y'=f(x,y)=x+y$, $x_o=0$, $y_o=1$, and final x=1.

Problem 2. Let $y'=f(x,y)=\exp(-y/x)+y/x$, $x_o=\exp(1)$, $y_o=0$, and final $x=1+\exp(1)$.

Investigation Mode (Optional). You may use any second order Runge-Kutta method to solve an initial value problem $y'=f(x,y)$. You must specify

1. $y'=f(x,y)$
2. initial conditions x_o, y_o and final x
3. parameters A, B, C, and D to satisfy A+B=1, B*C=B*D=1/2
4. the number of integration steps $N< 101$

Student Performance

Given $y'=f(x,y)$ with initial conditions x_o, $y(x_o)$, the student should be able to formulate any Runge-Kutta method by specifying the formulas for K1, K2, and $y_{i+1}=y_i$+A*K1+B*K2. The student should know what values of A,B,C, and D to use for the modified Euler's method and the improved Euler's method.

Lesson 23: Numerical Integration, Error Estimation, and Extrapolation

Reading Assignment

1. Read Conte, pp. 217-220.
2. Review Taylor's algorithm of order 2.
3. Review second order Runge-Kutta methods.
4. Review the mean value theorem for derivatives (see Conte, p.15).
5. Review the definition of the order of an approximation (see Conte, p. 115, formula 4.18).

Computer Assignment

Use the following section names to begin the two available modes:

1. L23L01 for the tutorial mode
2. L23P01 for the problem-investigation mode

Problem-Investigation Mode (Optional). You may apply the Taylor algorithm of order 2 or any second order Runge-Kutta method to approximate the solution to $y'=f(x,y)$ of your own choice. You must supply

1. $y'=f(x,y)$,
2. f_x and f_y in case of Taylor's algorithm,
3. A, B, C, and D in case of a Runge-Kutta method,
4. initial conditions x_o, y_o and final value for x, and
5. the number of integration steps $N < 101$.

The computer will provide the numerical integration results, ZN for N steps, Z2N for 2*N steps, and the extrapolated result Z=(4*Z2N-ZN)/3.

Student Performance

The student should know the general formulas for second order Runge-Kutta methods or Taylor's algorithm. On the basis of N and 2*N integration steps, the student should be able to compare the values of ZN and Z2N to determine a lower bound on the number of correct digits in the answer. Similarly, the student should compare Z2N and the extrapolated value Z.

Index

Titles of Lessons and Sections - Computer Section Names

Lesson A: Keyboard Orientation

1. Transfer of Control Between the Student and Computer L0L01
2. Correction of Typing Errors - the RUBOUT Key L0L02
3. Correction of Typing Errors - the # Key L0L03
4. Mathematical Expressions L0L04
5. Subscripted Variables L0L05
6. The Distinguished Name PI L0L06
7. Available Mathematical Functions - Latitude in Usage L0L07
8. First Estimates of Zeros (Roots) of Functions L0L08

Lesson B: Computer Numbers and Computational Error

1. Floating Point Representation L0L11
2. k-Digit Normalized Floating Point Representation L0L12
3. Errors in Computer Representation of Numbers L0L13
4. Errors Introduced by Computer Operations L0L14
5. Propogation of Error L0L15
6. Problem-Investigation Mode L0P01

Lesson 1: Linear Iteration - Methodology

1. Fixed Point of a Function L1L01
2. Geometric Meaning of a Real Fixed Point L1L02
3. Transformation of the Function $F(x)=0$ to the Equation $x=G(x)$ L1L03
4. Method of Iteration to Find a Zero of $F(x)$ L1L04
5. Sufficient Conditions for Convergence of an Iteration $x_{k+1}=G(x_k)$ L1L05

6. Tests for Convergence	L1L06
7. Problem Mode	L1P01
8. Investigation Mode	L1I01
Lesson 2: Linear Iteration - Theory	
1. Review of the Method	L2L01
2. Linear Iteration Theorem	L2L02
3. Meaning of Linear Iteration (Linear Convergence)	L2L03
4. Problem Mode	L2P01
5. Investigation Mode	L2I01
Lesson 3: An Acceleration Technique	
1. Geometric Sequences	L3L01
2. Aitken's Delta-Squared Process	L3L02
3. Aitken's Delta-Squared Process Applied to Linear Iteration	L3L03
4. Problem Mode	L3P01
5. Investigation Mode	L3I01
Lesson 4. Newton's Method and Quadratic Convergence	
1. Newton's Method	L4L01
2. Convergence Proof for Newton's Method	L4L02
3. Quadratic Convergence for Newton's Method	L4L03
4. Problem Mode	L4P01
5. Investigation Mode	L4I01
Lesson 5: The Secant Method	
1. The Iteration Equation	L5L01
2. Convergence Behavior of the Secant Method	L5L02
3. Problem Mode	L5P01
4. Investigation Mode	L5I01

Lesson 6: Simultaneous Equations - Newton's Method

1.	Review of Partial Derivatives - Notation	L6L01
2.	Derivation of Newton's Method for Functions of Two Variables	L6L02
3.	Newton's Iteration Formulas	L6L03
4.	Quadratic Convergence of Newton's Method	L6L04
5.	Problem Mode	L6P01
6.	Investigation Mode	L6I01

Lesson 7: Polynomial Equations - Real Roots

1.	Evaluation of Polynomials by Nested Multiplication	L7L01
2.	Review - Division Algorithm for Polynomials	L7L02
3.	Formal Derivation of the Nested Multiplication Algorithm for Evaluation of a Polynomial	L7L03
4.	Evaluation of the Derivative of a Polynomial	L7L04
5.	Newton's Method for Polynomials	L7L05
6.	Problem Mode	L7P01
7.	Investigation Mode	L7I01

Lesson 8: Difficulties in Finding Roots of Polynomials

1.	Review of Newton's Method for Polynomials	L8L01
2.	Behavior of Newton's Method for Double Roots	L8L02
3.	The Concept of Instability	L8L03
4.	Problem Mode	L8P01
5.	Investigation Mode	L8I01

Lesson 9: Recursion Formulas for Dividing a Polynomial by a Quadratic Factor and Review of Complex Arithmetic

1.	Division Algorithm	L9L01

2. An Algorithm for Computing the Coefficients of the Quotient Polynomial $Q(x)$ and Remainder $R(x)$ when Dividing a Polynomial $P(x)$ by a Quadratic Polynomial $x^2-S*x-T$	L9L02
3. Complex Numbers, Complex Conjugates, and Quadratic Polynomials with Complex Roots	L9L03
4. Problem Mode	L9P01
Lesson 10: The Newton-Bairstow Method for Finding Complex Zeros of a Polynomial	
1. Review of Division Algorithm for Dividing $p(x)=a_nx^n+a_{n-1}x^{n-1}+\ldots+a_o$ by x^2-Sx-T	L10L01
2. The Newton-Bairstow for Improving an Approximate Quadratic Factor	L10L02
3. Problem Mode	L10P01
4. Investigation Mode	L10I01
Lesson 11: The Solution of Linear Systems by Elimination	
1. Representation of Linear Systems by the Augmented Matrix	L11L01
2. Gaussian Elimination	L11L02
3. Formation of Gaussian Multipliers	L11L03
4. Problem Mode	L11P01
5. Investigation Mode	L11I01
Lesson 12: Evaluation of Determinants and Matrix Inversion	
1. Evaluation of Determinants by Using Gaussian Elimination	L12L01
2. Review of Matrix Multiplication (Optional)	L12L02
3. Matrix Inversion by Using Gaussian Elimination	L12L03
4. Use of the Inverse Matrix to Solve Linear Systems	L12L04
5. Problem-Investigation Mode	L12P01

Lesson 13: Errors and Conditioning

1. Ill-Conditioned Systems	L13L01
2. "Normalized Determinant" as a Measure of the Condition of the Coefficient Matrix	L13L02
3. Computation of Norm $\lvert A \rvert$ for the General N by N Matrix	L13L03
4. Iterative Process to Improve the Numerical Solution of an Ill-Conditioned System	L13L04
5. Problem-Investigation Mode	L13P01

Lesson 14: Iterative Methods for Solution of Linear Systems

1. Method of Simultaneous Displacements	L14L01
2. Matrix Formulation of the Method of Simultaneous Displacements	L14L02
3. Method of Successive Displacements	L14L03
4. Criteria for Terminating an Iteration	L14L04
5. Problem Mode	L14P01
6. Investigation Mode	L14I01

Lesson 15: Convergence of Iterative Methods for Linear Systems

1. Review - Method of Simultaneous Displacements	L15L01
2. Formation of the Error Equations	L15L02
3. Column Sum Criteria - Sufficient Conditions for Convergence of the Method of Simultaneous Displacements	L15L03
4. Row Sum Criteria - Sufficient Conditions for Convergence	L15L04
5. Dominance Tests for Convergence and Review of Lesson	L15L05
6. Problem Mode	L15P01
7. Investigation Mode	L15I01

Lesson 16: Numerical Differentiation

1. Order of an Approximation L16L01
2. Functions Tabulated on an Equally Spaced Set of Points L16L02
3. Numerical Approximation of $f'(x_i)$ L16L03
    a. $O(h)$-Approximation of $f'(x_i)$
    b. $O(h^2)$-Approximation of $f'(x_i)$ for $0<i<N$
    c. $O(h^2)$-Approximation of $f'(x_0)$ and $f'(x_N)$
4. $O(h^2)$-Approximation of $f''(x_i)$ for $0<i<N$ L16L04
5. Computational Accuracy of Numerical Differentiation L16L05
6. Problem Mode L16P01
7. Investigation Mode L16I01

Lesson 17: Extrapolation to the Limit

1. Review of the Order of an Approximation L17L01
2. Simple Extrapolation for Differentiation L17L02
3. Repeated Extrapolation for Differentiation L17L03
4. Extrapolation to the Limit for Differentiation L17L04
5. Problem Mode L17P01
6. Investigation Mode L17I01

Lesson 18: Numerical Integration-The Trapezoidal Rule

1. Notation for the Integral L18L01
2. Second Theorem of the Mean for Integrals L18L02
3. Review of Rolle's Theorem L18L03
4. Error in Linear Approximation of a Function L18L04
5. Derivation of the Trapezoidal Rule and Its Error L18L05
6. General Application of the Trapezoidal Rule L18L06

7. Problem Mode	L18P01
8. Investigation Mode	L18I01
Lesson 19: Romberg Integration	
1. Introduction	L19L01
2. Basic Differentiation Formulas	L19L02
3. General Formulation of the Trapezoidal Rule	L19L03
4. Romberg Integration-Simple Extrapolation	L19L04
5. Romberg Integration-Repeated Extrapolation	L19L05
6. Romberg Integration-Extrapolation to the Limit	L19L06
7. Problem Mode	L19P01
8. Investigation Mode	L19I01
Lesson 20: Numerical Integration-Simpson's Rule	
1. Review of the Trapezoidal Rule	L20L01
2. Review of Romberg Integration - Simple Extrapolation	L20L02
3. Simpson's Rule on an Interval of Length 2h	L20L03
4. General Form of Simpson's Rule	L20L04
5. Error Formula for Simpson's Rule	L20L05
6. Problem Mode	L20P01
7. Investigation Mode	L20I01
Lesson 21: Numerical Integration of Ordinary Differential Equations by Taylor Series Approximation	
1. Statement of the Initial Value Problem	L21L01
2. Taylor's Algorithm of Order 1 - Euler's Method	L21L02
3. Review of Notation for Partial and Total Derivatives	L21L03
4. Taylor's Algorithm of Order 2	L21L04
5. Taylor's Algorithm of Order 3	L21L05

6. Taylor's Algorithm of Order k L21L06

7. Problem Mode L21P01

8. Investigation Mode L21I01

Lesson 22: Second Order Runge-Kutta Methods

1. Introduction L22L01

2. Numerical Example of a Second Order Runge-Kutta Method L22L02

3. Optimal Parameters A, B, C, and D L22L03

4. Special Cases and a Look at the Local Error L22L04

5. Problem Mode L22P01

6. Investigation Mode L22I01

Lesson 23: Numerical Integration, Error Estimation, and Extrapolation

1. Review of Second Order Methods for Solution of $y'=f(x,y)$ L23L01

2. Estimation of the Cumulative Error $y_N-y(x_N)$ L23L02

3. Practical Estimation of $y_N-y(x_N)$ and Extrapolation L23L03

4. Problem-Investigation Mode L23P01

APPENDIX B

LESSON PLANS

In order to provide the reader with a description of the topics presented in the course, the depth of each individual topic, and the degree of student participation, an outline of each lesson is presented in a condensed form. Throughout the outline, the general activities of the student are either explicitly stated or take the form of an exercise. All other activities amount to a computer presentation of course material with no student interaction. No description of the problem and investigation modes is presented here since these modes are described in the associated study guides in the Student Manual (see Appendix A). The reader may find it useful to further consult the Student Manual for reading assignments and the expected student performance.

Lesson A: Keyboard Orientation

Purpose

Introduce those keys on the KSR-33 Teletype terminal which will be frequently used by the student in constructing responses. Define the arithmetic operators, function names, and other symbols which can be used in constructing mathematical responses.

Prerequisites

The student needs an elementary knowledge of arithmetic expressions in Fortran.

Lesson Outline

1. Communication Between the Student and Computer. Demonstrate the use of the (RETURN) key in signaling completion of a response and the use of the # character to signal the expectation of a response.

Exercises: The student practices using the (RETURN) key.

2. Correction of Typing Errors--the (RUBOUT) Key. Demonstrate the use of the (RUBOUT) key as a logical eraser.

Exercises: The student practices the correction of typing errors with the (RUBOUT) key.

3. Correction of Typing Errors--the # Key. Demonstrate the use of the # key by the student as a logical back space.

Exercises: The student practices using the # key and (RUBOUT) key.

4. Arithmetic Operators. Describe the keyboard location of the +, -, *, and / keys.

Exercises. The student practices typing several arithmetic expressions.

5. Subscripted Variables. Describe the use of [and] as delimiters of subscript expressions.

Exercises: The student types subscripted expressions.

6. The Distinguished Name PI. Define PI as the name for the transcendental number π.

Exercise: What are three distinct values of x (in radians) so that sin(x)=0?

7. Available Mathematical Functions. List the names of functions which are available to the student, e.g. trigonometric, exponential, logarithm, etc.

8. First Estimates of Zeros of Functions. Describe the first method as the location of an interval [A,B] on which $f(x)$ is continuous and $f(A)f(B)<0$.

Exercise: Name an interval [A,B] on which $f(x)$ is continuous and $f(A)f(B)<0$ for $f(x)=x-e^{-x}$, for $f(x)=x+2-e^{-x}$, for $f(x)=2x-\tan(x)$.

The student may repeat the last exercise for any of the $f(x)$ as often as he wishes.

Describe the second method as the transformation of $f(x)=0$ into the form $h(x)=g(x)$ where, in the latter form, one seeks a point where h and g have the same value. Demonstrate the concept for $f(x)=x-e^{-x}$ by letting $h(x)=x$ and $g(x)=e^{-x}$.

Exercise: Let $f(x)=x-\cos(x)$. Name two functions $g(x)$ and $h(x)$ which have a common point at a zero of $f(x)$.

Exercise: Same as previous exercise with $f(x)=x^3+x^2-2x-2$.

The student may repeat the last two exercises as often as he wishes.

Describe the third method as one which applies for real roots of polynomials. For $p(x)=a_nx^n+\ldots+a_o$, the solution to $q(x)=a_nx+a_{n-1}$ may give a "good" estimate of a root which is relatively large in magnitude. The solution to $q(x)=a_1x+a_o$ may yield a "good" estimate of a relatively small root.

Exercise: Estimate the largest and smallest root (in magnitude) of $p(x)=x^3-11.1x^2+11.1-1$.

Exercise: Estimate the largest root of $p(x)=x^4+10x^3-11.2x^2-2x+2.2$.

Lesson B: Computer Numbers and Computational Error

Purpose

Introduce a format for floating point representation of numbers in terms of an exponent and mantissa. Demonstrate the finiteness of a computer by counting the available floating point numbers. Demonstrate the effects of truncation, symmetric rounding, propogation of error, and loss of significance. Illustrate possible instability in recursion formulas.

Prerequisites

The student must be familiar with the terminal keyboard (see Lesson A).

Lesson Outline

1. Floating Point Representation (E-format). Define the floating point representation and the function of the mantissa and exponent.

Exercise 1A: The student must select illegal representations from a list of possible floating point representations.

2. k-Digit Normalized Floating Point Representation. Define a k-digit normalized floating point form as a normalized floating point number with a k-digit mantissa. If the exponent is zero, then the exponent carries a positive sign.

Exercise 2A: Write N=-.04387 in 5-digit normalized form.

3. Errors in Computer Representation of Numbers.

Exercise: How many "positive" 5-digit normalized numbers can be represented by .XXXXXEPYY where P=+ or -? How many "negative" 5-digit normalized numbers can be represented by the form -.XXXXXEPYY?

Define symmetric rounding of a number.

Exercise 3A: What is the 4-digit rounded computer representation of N=-10/7?

Define truncation of excessive digits without rounding.

Exercise 3B: What is the 4-digit truncated computer representation of N=10/7? of N=-1/3?

4. Errors Introduced by Computer Operations. Demonstrate the 4-digit normalized result (x+y)'=.1013E+02 when x=.6717E+01 and y=.3412E+01.

Exercise 4A: Write the 4-digit normalized floating point representation of (x+y)' when x=.3675 and y=.8734.

Define loss of significance.

Exercise 4B: Let x=.36214743 and y=.436173111. In a 5-digit machine, x'=.43621E+00 and y'=.43617E+00. x' and y' agree with x and y through how many digits? What is the 5-digit computer result (x'-y')'? The exact result is x-y=$.41632\cdot 10^{4}$. (x'-y')' agrees with x-y through how many digits? Let z=.321745823 with the computer representation z'=.32175E+00. Assuming the double precision dividend z' and the single precision divisor (x'-y')'=.40000E-04, what is the 5-digit rounded normalized representation (z'/(x'-y')')'?

5. Propogation of Error. Describe two methods for computing n! times the remainder after n terms of the MacClaurin series for e^{x}.

Method 1: F(n, x)=n! (x^{n+1}/(n+1)!+...) where the process is terminated when n!(x^{p}/p!) becomes insignificant.

Method 2: F(0,x)=EXP(x)-1 where EXP(x) is computed as accurately as possible by the MacClaurin series and F(n+1,x)=(n+1)(F(n,x))-x^{n+1} for n=0,1....

Exercise 5A: The student may specify various values of x in the interval [-1,1] and observe the growth of the difference in the values computed by the two methods.

Lesson 1: Linear Iteration--Methodology

Purpose

Define various techniques for transforming the equation $f(x)=0$ to a form $x=g(x)$ where $f(P)=0$ implies $P=g(P)$. Illustrate the importance of $g(x)$ through both convergent and divergent iterations $x_{k+1}=g(x_k)$. State sufficient conditions for convergence. Describe several decision techniques for stopping a convergent iteration.

Prerequisites

Recommended prerequisite material includes techniques for obtaining first estimates of zeros of functions (Lesson A) and propogation of round-off error (Lesson B).

Lesson Outline

1. Fixed Point of a Function. Define a fixed point of $g(x)$ as a value P so that $g(P)=P$.

Exercise 1A: Find a fixed point of $g(x)=2x-7$, $g(x)=x^2+x-1$, and $g(x)=x^3-B^3+x$.

2. Geometric Meaning of a Fixed Point. Construct a graph showing the relationship between a zero of the function $g(x)-x$ and the intersection of $y=x$ and $y=g(x)$.

Exercise 2A: $P=-1$ is a fixed point of $g(x)=x^2+x-1$. So P is a zero of what function?

Exercise 2B: Let $g(x)=2x-\tan(x)$ and suppose P is a number so that $g(P)=P$. Then P is the intersection between two curves, namely $y=x$ and $y=$?

Exercise 2C: How many "real" fixed points does $g(x)=x^2+x+1$ have?

3. Transformation of $f(x)=0$ to $x=g(x)$. Demonstrate two methods for such a transformation. For method 1, set $f(x)=0$ and add x to both sides to obtain $x=g(x)=x+f(x)$.

Exercise: Compute $g(x)$, $g(-1)$, and $g(2)$ for $f(x)=x^2-x-2$.

For method 2, set $f(x)=0$, divide both sides by x, and subtract both sides from x.

Exercise 3A: Compute $g(x)$, $g(-1)$, and $g(2)$ for $f(x)=x^2-x-2$.

Exercise 3B: Suppose $f(P)=0$, $P>0$. For which case is $g(P)\neq P$?

Case 1: $g(x)=x+f(x)/c$ where $c\neq 0$

Case 2: $g(x)=\mathrm{SQRT}(x^2-f(x))$

Case 3: $g(x)=-\mathrm{SQRT}(x^2+f(x))$

4. Method of Iteration to Find a Zero of $f(x)$. Describe the iterative process $x_{k+1}=g(x_k)$.

Exercise 4A: For $f(x)=x^2-x-2$, one possibility is $g(x)=x-f(x)/x$. Compute $g(x)$ in terms of x. If $x_o=5/4$, what is x_1 and x_2? Computation for succeeding iterations is provided until the student is convinced of convergence.

Tell the student that the choice of both the iteration function and the starting value may be crucial for convergence.

Exercise 4B: For $f(x)=x^2-x-2$, choose the iteration function $g(x)=x+f(x)=x^2-2$. The student selects $x_o>2$ and observes successive iterations. The student exits from the iteration cycle when convinced of divergence. This exercise may be repeated as often as the student wishes.

5. Sufficient Conditions for Convergence of an Iteration $x_{k+1}=g(x_k)$.

$x_{k+1}=g(x_k)$ will converge if x_o is appropriately chosen in an interval (A,B) so that

a. $A<P<B$

b. $g(P)=P$

c. $g(x)$ and $g'(x)$ are continuous on (A,B)

d. $|g'(x)| < 1$ on (A,B)

Exercise 5A: Check the conditions for convergence if $f(x)=2x-e^{-x}$, $A=0$, $B=1$, and $g(x)=x-f(x)/3$.

Exercise 5B: Let $f(x)=2x-e^{-x}$, $A=0$, $B=1$, and $g(x)=(x-f(x))/3$. Which condition is not satisfied?

6. Tests for Convergence. Explain the use of the absolute error test $|x_{k+1}-x_k| < \epsilon$ and the relative error test $|(x_{k+1}-x_k)/x_{k+1}| < \epsilon$. The student is given three sequences of iterates and asked to determine the value of k which satisfies the two error tests.

Lesson 2: Linear Iteration--Theory

Purpose

Establish and verify sufficient conditions for convergence of an iteration $x_{k+1}=g(x_k)$ (see Conte, Theorem 2.1). Define linear convergence as $\lim e_{k+1}/e_k \rightarrow$ constant where $e_k=x_k-P$. Establish linear convergence when Theorem 2.1 is satisfied.

Lesson Outline

1. Review of the Method:

Exercise 1A: Let $f(x)=x-\cos(x)/3$ with a zero P in the interval (0,PI/2). Check the following conditions for $g(x)=x-f(x)$:

a. $g(P)=P$

b. continuity of $g(x)$ and $g'(x)$ on (0,PI/2)

c. $|g'(x)| < 1$ on (0,PI/2)

2. Linear Iteration Theorem. State the theorem (see Conte, Theorem 2.1).

The student participates in the proof of this theorem through multiple choice type responses. The mean-value theorem is stated if the student has difficulty in applying it during the proof.

Exercise 2A: Let $f(x)=x-e^{-x}$ and $g(x)=x-f(x)$. The student must check each of the conditions for convergence if x_0 is chosen in (.3,.75).

3. Meaning of Linear Iteration. Form the error equation $e_k=x_k-P$.

The student participates in the derivation of $\lim_{k\to\infty} \frac{e_{k+1}}{e_k} = g'(P)$ through multiple choice type items.

Exercise 3A: Let $g(x)=(x^2-f(x))^{\frac{1}{2}}$ with $f(x)=x^2-e^{1-x}$ and P=1. The student checks all conditions to assure convergence of $x_{k+1}=g(x_k)$ for $x_0\epsilon(0,2)$. The student chooses any x_0 in (0,2) different from 1 and requests successive iterations to observe the values $x_k, e_k, e_k/e_{k-1}$. The student exits from the iteration cycle when he is convinced that $x_k\to 1$, $e_k\to 0$, and $e_k/e_{k-1}\to g'(1)$.

Lesson 3: An Acceleration Technique

Purpose

Define Aitken's δ^2-formula. Demonstrate and then prove that Aitken's formula provides the exact limit of a geometric sequence. Relate geometric sequences to the error sequence $e_k=x_k-P$ for a linearly convergent iteration $\{x_k\}$. Demonstrate the asymptotic behavior of $\{e_k\}$ as a geometric sequence, thereby making Aitken's formula an acceleration technique.

Prerequisites

It is essential that the student be familiar with the material in Lessons 1 and 2, in particular, the sufficiency conditions for convergence and the meaning of "linear convergence".

Lesson Outline

1. Geometric Sequences. Define a geometric sequence $e_{k+1}=Me_k$.

Exercise 1A: What is the value for M in the geometric sequence $\{2,3/2,9/8,27/32,\ldots\}$? Compute e_4. Compute $\lim_{k\to\infty} e_k$.

Exercise 1B: Compute M for the geometric sequence $\{3,-6/5,12/25,-24/125,\ldots\}$. Compute $\lim_{k\to\infty} e_k$.

Exercise 1C: Suppose $x_k=P+e_k$ and $e_k=Me_{k-1}$ with $-1<M<1$. Compute $\lim_{k\to\infty} x_k$.

2. Aitken's δ^2-Process. Define Aitken's δ^2-Process as $x_k'=x_{k-2}-(x_{k-1}-x_{k-2})^2/(x_k-2x_{k-1}+x_{k-2})$.

Exercise 2A: Consider the sequence $\{x_0,x_1,\ldots\}$ given by $\{3,5/2,17/8,\ldots\}$. Compute x_2' and x_3'.

Point out that the sequence in Exercise 2A can be written as $\{1+2, 1+3/2, 1+9/8, \ldots\}$ where $\{2, 3/2, 9/8, \ldots\}$ is a geometric sequence with $M=3/4$.

Exercise: Compute $\lim_{k \to \infty} x_k$ for the sequence in Exercise 2A.

Exercise 2B: Consider the sequence $\{x_k\}$ given by $\{3.01, 3.001, 3.0001, 3.00001, \ldots\}$. Use Aitken's formula to compute x_k' for any $k>1$.

Exercise: The sequence in Exercise 2B can be written as $\{3+e_0, 3+e_1, 3+e_2, \ldots\}$ where $\{e_k\}$ is a geometric sequence. Compute M, $\lim e_k$, and $\lim x_k$.

Formally prove that Aitken's δ^2-process will give the exact limit of the sequence $\{x_0, x_1, x_2, \ldots\}$ where $x_k=P+e_k$ and $\{e_k\}$ is a geometric sequence with $-1<M<1$. The student participates in this proof through both constructed responses and multiple choice type items.

3. Aitken's δ^2-Process Applied to Linear Iteration. Point out to the student that the sequence of errors $\{e_k\}$ is asymptotically geometric for a lineraly convergent iteration $\{x_k\}$.

Exercise 3A. Let $f(x)=x^2-x-2$, $g(x)=x-f(x)/x=1+2/x$, and $x_o=1.5$. The student progresses through the computation on linear iteration by observing the values x_o, x_1, x_2, $x_2'=x_3$, x_4, x_5, $x_5'=x_6$, etc. The student exits from the iteration cycle when he feels he has observed the effect of acceleration.

Exercise 3B: The student progresses through the linear iteration in Exercise 3A without acceleration and observes the values x_k, e_k, and e_k/e_{k-1}. The student is asked to observe convergence of x_k to 2, e_k to 0, and e_k/e_{k-1} to $g'(2)=-\frac{1}{2}$, noting that in the latter iterations

$e_k \approx (-\frac{1}{2})e_{k-1}$, i.e. approximately geometric.

Lesson 4: Newton's Method and Quadratic Convergence

Purpose

Define Newton's iteration by $x_{k+1}=x_k-f(x_k)/f'(x_k)$. State the continuity conditions of f, f', and f" which will insure a convergent iteration for the proper choice of x_o. Prove that these conditions are sufficient for convergence by using the conditions in the Linear Iteration theorem (see Lesson 2). Define quadratic convergence by $e_{k+1}/e_k^2 \rightarrow$ constant. Prove that Newton's iteration converges quadratically with $e_{k+1}/e_k^2 \rightarrow g''(P)/2$.

Prequisites

The student is expected to know the Linear Iteration theorem and the meaning of linear convergence from Lesson 2.

Lesson Outline

1. Newton's Method. Define the iteration function as $g(x)=x-f(x)/f'(x)$.

Exercise 1A: Write Newton's iteration function for $f(x)=x-3\sin(x)$.

Exercise 2A: Write Newton's iteration function for $f(x)=\sin(x)\cos(x)$.

Exercise 3A: Let $f(x)=x^2-3$. Write Newton's iteration function g(x), compute g(SQRT(3)), and write the iteration equation $x_{k+1}=(?)$. The student then chooses a starting value $1<x_o<2$ such that $x_o\neq$SQRT(3) and is asked to observe that the number of correct decimal places approximately doubles with each iteration. The student exits from the iteration cycle when he is satisfied.

In preparation for a convergence proof, the student has the option of reviewing a statement of the Linear Iteration theorem.

2. Convergence Proof for Newton's Method. State the overall assumptions that P is a simple zero of $f(x)$; f, f', and f'' are continuous on (A,B) where $A<P<B$; and $g(x)=x-f(x)/f'(x)$ is the iteration function. Through multiple choice items, the student participates in the proof that $g(x)$ satisfies the Linear Iteration theorem for some symmetric interval about P.

Exercise 2A: $f(x)=x^2-.01x$ has a zero at $P=0$. Name an interval $(A1,B1)$ on which f, f', and f'' are continuous and $A1<P<B1$. Name an interval $(A2,B2)$ contained in $(A1,B1)$ on which g and g' are continuous and $A2<P<B2$. Name an interval (A,B) contained in $(A2,B2)$ so that $A<P<B$ and on which $|g'(x)| < 1$.

3. Quadratic Convergence of Newton's Method. The general assumptions given in Section 2 are restated. The student is asked to recall the error equation $e_{k+1}=g'(z_k)e_k$.

Exercise: To establish linear convergence, the student is asked to compute $\lim(e_{k+1}/e_k)$. Through multiple choice items, the student participates in the proof that $\lim(e_{k+1}/e_k^2)=g''(P)/2$, establishing quadratic convergence.

Exercise 3A: For $f(x)=xe^x$, write Newton's $g(x)$, $g'(x)$, $g'(P)$ where $P=0$, and an interval (A,B) so that $A<P<B$ and on which $|g'(x)| < 1$. The student may select any x_o from (A,B) and observe the values for x_k, e_k, e_k/e_{k-1}, and e_k/e_{k-1}^2 for $k=1,2,\ldots.$ During the iteration, the student is asked to observe $e_k/e_{k-1}\to 0$ and $e_k/e_{k-1}^2\to g''(P)/2$. The student may exit from the iteration cycle when satisfied.

Lesson 5: The Secant Method

Purpose

Define the secant method. Demonstrate by examples that $e_k \to 0$, $e_k/e_{k-1} \to 0$, $e_k/(e_{k-1}e_{k-2}) \to$ constant and $e_k/e_{k-1}^2 \to \infty$ when f, f', and f" are continuous, thereby establishing the convergence rate as better than linear but not as good as quadratic.

Prerequisites

Knowledge of Newton's method, linear convergence, and quadratic convergence is assumed.

Lesson Outline

1. The Iteration Equation. Define the secant method by replacing $f'(x_k)$ in Newton's iteration by an approximation to obtain $x_{k+1}=x_k-f(x_k)(x_k-x_{k-1})/(f(x_k)-f(x_{k-1}))$.

Exercise 1A; Let $f(x)=x^2-x-2$. Write the secant method.

Define the simplified form $x_{k+1}=(x_{k-1}f(x_k)-x_kf(x_{k-1}))/(f(x_k)-f(x_{k-1}))$.

Exercise 1B: Write the secant method when $f(x)=\cos(x)$.

2. Convergence of the Secant Method.

Exercise 2A: $f(x)=1+1/x$ has a simple zero at $P=-1$. Write the secant method. If $x_0=-.5$ and $x_1=1.5$, compute x_2. Computation for succeeding iterations provides the values of x_k, e_k, e_k/e_{k-1}, e_k/e_{k-1}^2, and $e_k/(e_{k-1}e_{k-2})$. The student observes that $e_k \to 0$, $e_k/e_{k-1} \to 0$, $e_k/e_{k-1}^2 \to \infty$, and $e_k/(e_{k-1}e_{k-2}) \to R=$constant. The student exits from the iteration cycle when he is satisfied.

Exercise: For Exercise 2A, compute $f''(P)/(2f'(P))$ and compare this value with $R=\lim(e_k/(e_{k-1}e_{k-2}))$.

The student is told that R is the rate of convergence of the secant method when f, f', and f" are continuous.

Exercise 2B: $f(x)=x^2-3$ has a simple zero at P=SQRT(3). Compute $\lim(e_k/(e_{k-1}e_{k-2}))$ and write the iteration equation for the secant method. The student selects values for $x_o > 0$ and $x_1 > 0$, $x_o \neq x_1$. Computation for succeeding iterations provides values for x_k, e_k, e_k/e_{k-1}, e_k/e_{k-1}^2, and $e_k/(e_{k-1}e_{k-2})$ to demonstrate convergence is better than linear, but not quadratic. The student may exit from the iteration cycle when $| x_{k+1}-x_k | < .5 \cdot 10^{-9}$.

Lesson 6: Simultaneous Equations--Newton's Method

Purpose

Derive Newton's method for two simultaneous equations in two variables. State sufficient conditions for quadratic convergence and illustrate both the method and convergence properties through examples.

Prerequisites

The student must be familiar with Taylor expansions for functions of two variables, Newton's method for functions of a single variable, and quadratic convergence.

Lesson Plan

1. Review of Partial Derivatives--Notation. State the definition of partial derivatives through the second order using a notation acceptable for the KSR-33 Teletype.

The student works two miscellaneous exercises for practice in computing partial derivatives.

Introduce a notation for a partial derivative evaluated at a point.

Exercise 1A: Let $f(x,y)=.1x^2+.1y^2+.8$ and $g(x,y)=.1x+.1xy^2+.8$. Write expressions in terms of x, y, x_o, and y_o for $f'_x(x_o,y_o)(x-x_o)+f'_y(x_o,y_o)(y-y_o)$ and $g'_x(x_o,y_o)(x-x_o)+g'_y(x_o,y_o)(y-y_o)$.

2. Derivation of Newton's Method for Functions of Two Variables. State that the motivation is to find a simultaneous solution of $f(x,y)=0$ and $g(x,y)=0$.

Exercise 2A: Find a simultaneous zero of $f(x,y)=(x-1)^2-y$ and $g(x,y)=(x-1)+y^2$.

Derive Newton's equations

$$f'_x(x_o,y_o)(x-x_o)+f'_y(x_o,y_o)(y-y_o)=-f(x_o,y_o) \text{ and}$$
$$g'_x(x_o,y_o)(x-x_o)+g'_y(x_o,y_o)(y-y_o)=-g(x_o,y_o)$$

using truncated Taylor series. The student participates through multiple choice type items.

3. Newton's Iteration Formulas. Derive the iteration formulas

$$x_{k+1}=x_k-(fg'_y-gf'_y)/J \text{ and}$$
$$y_{k+1}=y_k-(gf'_x-fg'_x)/J$$

where $J=f'_xg'_y-g'_xf'_y$. The student participates through multiple choice items.

Exercise 3A: $f(x,y)=x^2+y^2-1$ and $g(x,y)=xy$ are simultaneously zero at (0,-1). Write f'_x, f'_y, g'_x, g'_y, in terms of x and y. Write J in terms of x_k, y_k, and J. The student chooses starting values $-.25<x_o<.25$ and $-1.25<y_o<-.75$. Succeeding iterations are computed and the student is asked to observe the quadratic convergence. The student exits from the iteration cycle when satisfied.

4. Quadratic Convergence of Newton's Method. State sufficient conditions for quadratic convergence: f, g and all partials through the second order are continuous in a region R which contains the zero, $J\neq 0$ in R, and (x_o, y_o) is chosen sufficiently close to the simultaneous zero of f and g.

Exercise 4A: Check all conditions for quadratic convergence for f and g given in Exercise 3A.

Lesson 7: Polynomial Equations--Real Roots

Purpose

Define the nested multiplication algorithm to evaluate a polynomial, stress its economy, and illustrate how the algorithm can be restated in terms of recursion formulas. Formally, derive this method of evaluation and show that the last recursion formula $b_o=b_1z+a_o=p(z)$ is the same as the b_o in the division algorithm for polynomials $\frac{p(x)}{x-z} = q(x)+\frac{b_o}{x-z}$. Derive the result $p'(z)=q(z)=c_1$ where the recursion formulas may be used for q(x). Define Newton's method for polynomials as $x_{k+1}=x_k-b_o/c_1$.

Prerequisites

The student should be familiar with the division algorithm for polynomials from elementary algebra and Newton's method for functions of a single variable (see Lesson 4).

Lesson Outline

1. Evaluation of Polynomials by Nested Multiplication. Using a fourth degree polynomial p(x), demonstrate the nested form and the

recursion formulas to find $p(z)$: $p(x)=a_4x^4+\ldots+a_0=(((a_4x+a_3)x+a_2)x+a_1)x+a_0$, $b_4=a_4, b_3=b_4z+a_3,\ldots,b_0=b_1z+a_0$.

Exercise 1A: Let $p(x)=3x^4+4.5x^3-2.8x^2+1.1x-6$. Identify $a_4,\ldots,a_0$ and the recursions formulas for $b_4,\ldots,b_0$ needed to evaluate $p(z)$.

Point out the number of multiplications and additions needed to evaluate a polynomial by this method.

Exercise: For $z=2$, write the numeric values for $b_4,\ldots,b_0$ in Exercise 1A. Confirm $b_0=p(2)$.

Describe the general algorithm for an nth degree polynomial.

Exercise 1B: Let $p(x)=a_6x^6+a_5x^5+a_4x^4+a_2x^2+a_1x+a_0$. Note that $a_3=0$. Write the necessary recursion formulas $b_6,\ldots,b_0$ to find $p(z)$.

2. Review--Division Algorithm for Polynomials. If $p(x)$ is a polynomial of degree $n>1$, then $p(x)/(x-z)=q(x)+b_0/(x-z)$ where $q(x)$ is a polynomial of degree $n-1$ and b_0 is the remainder.

Exercise 2A: Let $p(x)=x^2-3x-4$ and $z=-1$. Write $q(x)$ and compute b_0.

Exercise 2B: Let $p(x)=x^3-x^2-x+1$ and $z=2$. Write $q(x)$ and b_0. Compute $p(z)$.

3. Formal Derivation of the Nested Multiplication Algorithm. Formally prove that $b_0=p(z)$ by substituting $q(x)=b_nx^{n-1}+\ldots+b_1$ into $p(x)=q(x)(x-z)+b_0$ where $p(x)=a_nx^n+\ldots+a_1$. The student participates by equating coefficients and solving for the b_1.

4. Evaluation of the Derivative $p'(z)$ of a Polynomial. Formally prove that $p'(z)=q(z)$ and demonstrate the use of recursion formulas to find this value. The student participates by writing the appropriate expressions for $c_n=b_n$, $c_{n-1}=c_nz+b_{n-1},\ldots,c_1=c_2z+b_1=p'(z)$.

5. Newton's Method for Polynomials. Define Newton's method using recursion formulas to find $b_0=p(x_k)$ and $c_1=p'(x_k)$.

Exercise 5A: Let $p(x)=x^3-6x^2+11x-6$ and $x_0=1.1$. Compute b_3, b_2, b_1, b_0, c_3, c_2, c_1, and x_1. Successive iterations are automatically provided. The student exits from the iteration cycle when satisfied.

Lesson 8: Difficulties in Finding Roots of Polynomials

Purpose

Review Newton's method for polynomials. Demonstrate the behavior of Newton's method in the case of double roots in terms of loss of significant digits. Demonstrate the concept of instability in polynomials of high degree.

Prerequisites

The student is expected to know Newton's method, linear and quadratic convergence (see Lessons 2 and 4), and Newton's method for polynomials.

Lesson Outline

1. Review of Newton's Method for Polynomials. Define the method.

Exercise 1A: Let $p(x)=x^3-3x^2+4$. Specify the coefficients a_i, the recursion formulas for the b_i, and the recursion formulas for the c_i.

2. Behavior of Newton's Method in the case of Double Roots. Point out that $p(x)$ in Exercise 1A has a double root at $x=2$. In other words $(x-2)^2$ is a divisor of $p(x)$.

Exercise: For $p(x)$ in Exercise 1A write an expression for $p'(x)$ and evaluate $p'(2)$. Write an expression for $p(x)/p'(x)$.

Point out that $p(x_k)/p'(x_k)\rightarrow 0$ in theory but significance is lost in actual computation of the ratio since $p'(x_k) \to 0$. Thus the convergence is linear and accuracy is poor.

Exercise: Starting with x_o=2.000001, in the last exercise, the student observes successive iterations, noting the linear convergence and the loss of significance beyond the sixth decimal place. The student must observe at least seven iterations and then may exit from the iteration cycle when satisfied.

3. The Concept of Instability. Define instability as the condition where small changes in the coefficients of p(x) produce large changes in the roots of p(x).

Exercise 3A: Let $p(x)=x^7-28x^6+322x^5-1960x^4+6769x^3-13132x^2+13068x-5040$. p(x) has exact roots at 1,2,3,...,7. The student selects a starting value $-.75<x_o<7.25$ and observes the convergence to a root until $|x_{k+1}-x_k| < .5\cdot10^{-8}$.

Exercise 3B: Rework Exercise 3A with $a_2=-13133$ and the same x_o. The student observes the difference in the roots of the two polynomials.

The student is given the option of repeating Exercises 3A and 3B as a group as often as desired.

Lesson 9: Recursion Formulas for Dividing a Polynomial by a Quadratic Factor and Review of Complex Arithmetic

Purpose

Review the algorithm for dividing a polynomial by a quadratic polynomial $p(x)/(x^2-Sx-T)=q(x)+r(x)/(x^2-Sx-T)$ where $\deg(p(x))=n\geq2$ implies $\deg(q(x))=n-2$ and $\deg(r(x))\leq1$. Derive an algorithm in terms of recursion formulas for computing the coefficients of q(x) and r(x). Review complex numbers. Prove that a quadratic polynomial $(x-z)(x-\bar{z})$

has real coefficients. All concepts in this lesson are preparatory for Lesson 10.

Prerequisites

The student should be familiar with the recursion formulas for dividing a polynomial by a linear factor (see Lesson 7).

Lesson Outline

1. Division Algorithm. Define the factorization of $p(x)$ as $p(x)=q(x)(x^2-Sx-T)+r(x)$ where $deg(p(x))=n>1$ implies $deg(q(x))=n-2$ and $deg(r(x))<2$.

Exercise 1A: Let $p(x)=x^4-2x^3+3x^2-2x+2$ and the quadratic divisor be x^2-2x+1. Compute S, T, $q(x)$, $r(x)$, $deg(q(x))$, and $deg(r(x))$.

Exercise 1B: Let $p(x)=x^5+x^4+x^3-x^2-x-2$ and the divisor be x^2+x+1. Compute S, T, $q(x)$, $r(x)$, $deg(q(x))$, and $deg(r(x))$.

Describe the general forms $p(x)=a_nx^n+\ldots+a_o$, $q(x)=b_nx^{n-2}+\ldots+b_2$, and $r(x)=b_1(x-S)+b_o$.

Exercise 1C: Compute $b_4,\ldots,b_0$ from the $q(x)$ and $r(x)$ in Exercise 1A.

Exercise 1D: Compute $b_5,\ldots,b_0$ from $q(x)$ and $r(x)$ in Exercise 1B.

2. An Algorithm for Computing the Coefficients of q(x) and r(x). Derive the recursion formulas for computing the b_i as $b_n=a_n$, $b_{n-1}=a_{n-1}+Sb_n$, $b_i=a_i+Sb_{i+1}+Tb_{i+2}$ for $i=n-2,\ldots,0$. The student participates by equating coefficients and supplying the right hand sides of the recursion formulas.

Exercise 2A: Let $p(x)=2x^5+x^4+3x^3+2x^2+x+3$ and the divisor be x^2-Sx-T. Compute $b_5,\ldots,b_0$ for the coefficients of $q(x)$ and $r(x)$ using the recursion formulas.

Exercise 2B: In Exercise 2A, let the divisor be x^2+2x+2. Compute S, T, q(x), and r(x).

3. Complex Numbers, Complex Conjugates, and Complex Roots. Define a complex number $z=u+vi$ with $Re(z)=u$ and $Imag(z)=v$.

Exercise 3A: Let $z=6.3+4.5i$. Compute Re(z) and Imag(z).

Define the complex conjugate of z by $\bar{z}=u-iv$.

Exercise 3B: For $z=6.3+4.5i$, compute $Re(\bar{z})$ and $Imag(\bar{z})$.

Exercise 3C: For $z=x+yi$, write Re(z), Imag(z), $Re(\bar{z})$, and $Imag(\bar{z})$.

Exercise: For $x^2-Sx-T=(x-z)(x-\bar{z})$, compute S and T. If $x=u+iv$, compute S and T in terms of u and v.

From the last exercise prove that, if quadratic polynomial x^2-Sx-T has complex roots z and $\bar{z}$, then S and T are real.

Exercise 3D: $p(x)=x^3-x^2+2$ has a complex root $z=1-i$. Name another complex root of p(x).

Point out that complex roots come in pairs for any polynomial with real coefficients.

Lesson 10: The Newton-Bairstow Method for Finding Complex Zeros of a Polynomial

Purpose

Describe the Newton-Bairstow method and its relationship to solving two simultaneous nonlinear equations in two variables. State Newton's equations which when solved will yield an approximate solution. Derive the recursion formulas for evaluating the partial derivatives in Newton's equations. Demonstrate the iterative process.

Prerequisites

The student is expected to know all concepts in Lesson 9, Newton's method for systems of equations (see Lesson 6), quadratic convergence, and partial derivatives.

Lesson Outline

1. Review of the Division Algorithm for Dividing by a Quadratic Polynomial. Define the recursion formulas.

Exercise 1A: Let $p(x)=x^4-2x^3+x^2+2x-2$ with an approximate root $z=1+.9i$. Write an approximate quadratic factor $x^2-S_ox-T_o$. Dividing $p(x)$ by this quadratic factor, compute $b_4,\dots,b_0$ for $q(x)$ and $r(x)$.

Point out that $r(x)\neq 0$ since the divisor was not exact and that the motivation will be to successively improve the divisor in order to annihilate $r(x)$.

2. The Newton-Bairstow Method for Improving a Quadratic Factor. Describe the problem by stating that $r(x)=b_1(x-S)+b_o\equiv 0$ is equivalent to finding S and T so that $b_1(S,T)=b_o(S,T)=0$. Using Newton's method in Lesson 6, an approximate solution for S and T can be found by solving Newton's equations

$$(b_1)'_S(S-S_o)+(b_1)'_T(T-T_o)=-b_1$$

$$(b_o)'_S(S-S_o)+(b_o)'_T(T-T_o)=-b_o$$

if we have a method for evaluating the partials at (S_o,T_o).

Exercise 2A: Let $p(x)=a_4x^4+\dots+a_o$. To compute $q(x)$ and $r(x)$, we use the recursion formulas $b_4=a_4$, $b_3=a_3+Sb_4$, $b_i=a_i+Sb_{i+1}+Tb_{i+2}$ for $i=2,1,0$. Compute $c_5=(b_4)'_S$. In terms of the b_i and any previously computed c_j, write $c_4=(b_3)'_S,\dots,c_1=(b_o)'_S$. Similarly compute

$d_{i+2}=(b_i)'_T$ for $i=4,\ldots,0$.

Demonstrate how the computation of the c_i and d_i form similar sets of recursion formulas and that actual computation of the d_i is not necessary. So Newton's system becomes

$$c_2(S-S_o)+c_3(T-T_o)=-b_1$$
$$c_1(S-S_o)+c_2(T-T_o)=-b_o.$$

Exercise 2B: $p(x)=x^4-20x^3+199x^2+20x-200$ has a complex zero near $9+10i$. Compute an approximate quadratic factor of $p(x)$, $x^2-S_ox-T_o$. Compute $b_4,\ldots,b_0$ for the coefficients of $q(x)$ and $r(x)$. Compute the values $c_4,\ldots,c_1$ for the partial derivatives. Solve Newton's system for improved values $S_1=S$ and $T_1=T$.

Describe the computational procedure for the Newton-Bairstow method in terms of the general iterates S_k and T_k.

Exercise: Automatic computation is provided for successive interations for Exercise 2B. The student exits from the iteration cycle when he is satisfied.

Lesson 11: The Solution of Linear Systems by Elimination

Purpose

Introduce Gaussian elimination as a systematic procedure for solving a linear system of equations. Describe the method of pivoting and demonstrate its usefulness as a control over the propagation of round-off error.

Prerequisites

This lesson is not dependent on any previous lesson.

1. Representation of Linear Systems by the Augmented Matrix. Demonstrate the formation of the augmented matrix $A|B$ for the linear system $Ax=b$.

Exercise 1A: The student forms the augmented matrix for a 3 x 3 linear system.

2. Gaussian Elimination. Describe the row operations:

a. (Row i) is replaced by (Row i)+M times (Row j)

b. Interchange (Row i) and (Row j)

Exercise 2A: For a 3 x 3 example, the student defines the values of i, j, and M for the first row operation to systematically generate zeros in positions a_{21}, a_{31}, and a_{32}. For each step, the student must specify new row values for the augmented matrix. The student solves the system by back-substitution.

3. Formation of Gaussian Multipliers. Present the student with a general 3 x 3 system of equations.

Exercise 3A: Write the formulas for the multipliers M to form zeros in the a_{21}, a_{31}, and a_{32} positions. Formally solve for x_3, x_2, and x_1 in the triangular system.

4. Gaussian Elimination with Pivoting. Justify the method of interchanging rows in order to avoid zero divisors.

Exercise 4A: The student performs the necessary interchanges in a 4 x 4 system.

Exercise 4B: The student uses Gaussian elimination with pivoting to solve a 3 x 3 system by specifying the necessary row operation listed in Section 2 along with the associated values of i, j, and M.

The student is asked to observe that the Gaussian multipliers never exceed the value 1 when pivoting is used, thereby lending stability to arithmetic process.

Lesson 12: Evaluation of Determinants and Matrix Inversion

Purpose

Demonstrate the use of Gaussian elimination in evaluating determinants and inverting matrices.

Prerequisites

Knowledge of all concepts in Lesson 11 is assumed.

Lesson Outline

1. Evaluation of Determinants by Using Gaussian Elimination.

Exercise: The student computes the determinant of a general 4 x 4 triangular matrix.

State that Gaussian elimination may be used to first reduce a matrix to triangular form and then the determinant is the product of its diagonal elements (with possible adjustment of the sign).

Exercise 1A: The student specifies the row operations needed to reduce a 3 x 3 system to triangular form. During the process the student requests one interchange of rows. The student computes the determinant of the intermediate matrices after each row operation and observes that the magnitude of the determinant is preserved but an interchange of rows changes the algebraic sign.

The general rules on sign changes are stated.

Exercise 1B: Suppose an n x n matrix A is reduced to the triangular matrix B such that det(B)=L. What is det(A) if three interchanges were required, if six interchanges were required, and if k interchanges were required?

2. Review of Matrix Multiplication. The student is given the option of omitting this section.

If the student elects to study this section, the method of multiplication is demonstrated for 3 x 3 matrices. The general formulas are stated.

Exercise 2A: The student forms the product of two 3 x 3 matrices.

Demonstrate multiplication of a 4 x 1 vector by a 4 x 4 matrix and state the general formulas

Exercise 2B: The student forms the product of a 4 x 4 matrix times a 4 x 1 vector.

3. Matrix Inversion by Using Gaussian Elimination. Define the inverse of a general n x n matrix as that n x n matrix A^{-1} so that $AA^{-1}=I$ where $I_{jk}=0$ for $j\neq k$ and $I_{jk}=1$ for $j=k$.

The student may optionally skip Exercise 3A.

Exercise 3A: The student multiplies two matrices, one of which is the inverse of the other.

Illustrate the general form of the coefficient matrix augmented by the identity matrix and describe the elimination process for simultaneous reduction of the n systems.

Exercise 3B: The student reduces a 3 x 3 matrix (augmented by the 3 x 3 identity matrix) to triangular form by specifying the necessary row operations. The student solves for the unknowns in each system by back-substitution to form the elements of the inverse matrix.

4. Use of the Inverse Matrix to Solve Linear Systems. Describe the method of solving the system $Ax=b$ by forming A^{-1} and solving $x=A^{-1}b$.

Exercise 4A: The student uses the inverse matrix A^{-1} from Exercise 3B to solve the system Ax=b for several b-vectors.

Point out the usefulness of this method for solving a set of linear systems $Ax=b_1$, $Ax=b_2,\ldots,Ax=b_n$.

5. Inversion of Ill-Conditioned Matrices.

Exercise 5A: The student solves a 3 x 3 ill-conditioned system of equations Ax=b and checks his approximate inverse B by observing that the elements of BA are in error in the seventh decimal place even though all computations were performed with fifteen digit precision.

Lesson 13: Errors and Conditioning

Purpose

Demonstrate the possible effects of propagation of round-off error and loss of significance in Gaussian elimination. Introduce several techniques for detecting an ill-conditioned linear system and describe possible remedial action.

Prerequisites

The student is expected to know all methods introduced in Lessons 11 and 12 involving Gaussian elimination with and without pivoting, computation of determinants, and inversion of matrices.

Lesson Outline

1. Ill-Conditioned Systems. Define an ill-conditioned system Ax=b as one in which small changes in the coefficients lead to large changes in the solution.

Exercise 1A: The student solves two systems Ax=b where in the first,

$$A = \begin{bmatrix} .99 & 1 \\ 1 & .99 \end{bmatrix},$$

and in the second,

$$A = \begin{bmatrix} 1 & 1 \\ 1 & .99 \end{bmatrix}.$$

For both cases b=column vector (1.99,1.99). The student observes that a change of 10^{-2} in a coefficient will give rise to a completely different solution.

Describe the method in Exercise 1A as one way to test for an ill-conditioned system.

2. Normalized Determinant as a Measure of the Coefficient Matrix. Describe the philosophy of a "normalized" determinant and state the computational formulas for a 2 x 2 system.

Exercise 2A: Compute the normalized determinant of

$$A = \begin{bmatrix} .99 & 1 \\ 1 & .99 \end{bmatrix}.$$

Exercise 2B: Compute the normalized determinant of

$$A = \begin{bmatrix} 3.1 & 4.2 \\ 6.2 & -1.8 \end{bmatrix}.$$

3. Computation of norm $|A|$ for the General n x n Matrix. Describe the method and state the computational formulas for norm $|A| = |A| / (\alpha_1 \ldots \alpha_n)$ where $\alpha_1 = \left(\sum_j a_{1j} \right)^{\frac{1}{2}}$.

Exercise 3A: The student computes norm | A | for

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

and notes that when the rows of A are mutually orthogonal, the system is "well-conditioned".

Exercise 3B: The student specifies the row operations needed to reduce

$$A = \begin{bmatrix} .24 & .36 & .12 \\ .16 & .20 & .26 \\ .12 & .16 & .24 \end{bmatrix}$$

to triangular form, and computes $|A|$, $\alpha_1, \alpha_2, \alpha_3$, and norm $|A|$.

The student is asked to observe a third indication of ill-conditioning in Exercise 3B, that is, the loss of one or more orders of magnitude in the pivotal elements during the reduction to triangular form.

4. Iterative Process to Improve the Numerical Solution of an Ill-Conditioned System. Define the residual vector as r=b-Ax' where x' is the numerical solution to Ax=b.

Exercise 4A: The student computes the residual vector for the case where

$$A = \begin{bmatrix} .99 & 1 \\ 1 & .99 \end{bmatrix}, \quad b = \begin{bmatrix} 1.99 \\ 1.99 \end{bmatrix}, \text{ and } \quad z = \begin{bmatrix} 1.99 \\ 0 \end{bmatrix}.$$

Derive the error system Ae=r where e=x-x' and show how a new approximation x"=x'+e can result in an improved solution.

Exercise 4B: For the system in Exercise 4A, the student solves the error system to find the approximate error e'. The student computes an improved solution $x''=x'+e'$.

Exercise 4C: The student is given the 3 x 3 system

$$A = \begin{bmatrix} -5.7958546 & 5.5934805 & 6.7754432 \\ -1.1022148 & 8.7379886 & 2.9751895 \\ -5.6710841 & 5.6330547 & 6.6816331 \end{bmatrix} \quad b= \begin{bmatrix} -47.346425 \\ -8.0130224 \\ -46.379222 \end{bmatrix}$$

The student specifies the row operations needed to reduce the augmented matrix to triangular form, computes $\alpha_1,\alpha_2,\alpha_3$, and norm $|A|$, computes the approximate solution x_3', x_2', x_1' by back-substitution, computes the residual vector $r'=b-Ax'$, solves the error system $Ae'=r'$ for e', and computes the improved solution $x_i''=x_i'+e_i'$ for $i=1$, 2, and 3.

Lesson 14: Iterative Methods for Solution of Linear Systems

Purpose

Describe the methods of simultaneous displacements and successive displacements. Illustrate how a reordering of the equations may be needed to insure nonzero diagonal elements.

Prerequisites

The student should be familiar with iterative methods for functions of a single variable (see Lesson 1).

Lesson Outline

1. Method of Simultaneous Displacements.

Exercise 1A: The student is given a 3 x 3 linear system $Ax=b$. He solves for x_i in the ith equation for $i=1,2,3$.

From the results of Exercise 1A, a subscript k is assigned to the variables appearing to the right of the "=" sign and a subscript k+1 to the variables appearing on the left. The student is asked to note that three iteration equations have been formed.

Exercise 1B: The student selects initial approximations $x_1^{(o)}$, $x_2^{(o)}$, and $x_3^{(o)}$. Computation for successive iterations $x_1^{(k)}$, $x_2^{(k)}$, and $x_3^{(k)}$ are provided. The student exits from the iteration cycle when he is convinced of convergence to the true solution of $Ax=b$.

Exercise 1C: The student forms the iteration equations for the method of simultaneous displacements for a general 3 x 3 system.

Describe the method of simultaneous displacements for the general n x n system.

2. Matrix Formulation of the Method of Simultaneous Displacements Describe the structure of the iteration matrix C where $x^{(k+1)}=Cx^{(k)}+d$ is the method of simultaneous displacements.

Exercise 2A: The student constructs the matrix C from a given 4 x 4 system of equations and observes that the diagonal elements are zero and the off-diagonal elements are of the form $c_{ij}=-a_{ij}/a_{ii}$. The student constructs the vector d and writes the iteration equations using the C-matrix and d-vector.

Exercise 2B: For a 3 x 3 system of equations with zero diagonal elements in the coefficient matrix, the student reorders the system so that the method of simultaneous displacements can be applied.

3. Method of Successive Displacements. The method of successive displacements is described for a 3 x 3 system.

Exercise 3A: The student constructs the iteration equations for a given 3 x 3 system.

Exercise 3B: The student repeats Exercise 3A for a different 3 x 3 system.

4. Criteria for Terminating an Iteration. Describe the absolute error test and relative error test.

Lesson 15: Convergence of Iterative Methods for Linear Systems

Purpose

Formulate the error equations for the method of simultaneous displacements and derive the column sum criteria for convergence based on the iteration matrix. Demonstrate how the row sum criteria for the iteration matrix may be used to establish convergence. Describe the relation of the row and column sum criteria of the iteration matrix to the diagonal dominance of the original coefficient matrix.

Prerequisites

The student must know both methods described in Lesson 14.

Lesson Outline

1. Review of the Method of Simultaneous Displacements.

Exercise 1A: Given a 3 x 3 system of equations $Ax=b$, the student constructs the iteration equations, the iteration matrix C, and the vector d for $x^{(k+1)}=Cx^{(k)}+d$.

2. Formation of the Error Equations.

Exercise 2A: The student constructs the iteration matrix C and vector d for a general 3 x 3 system.

For a general 3 x 3 system, formally derive the error equations for the error system $e^{(k+1)}=Ce^{(k)}$. The student participates through multiple choice items and constructed responses.

3. Column Sum Criteria as Sufficient Conditions for Convergence. Formally prove that if $\sum_i |c_{ij}| < 1$ for $j=1,\ldots,n$, then the method of simultaneous displacements will converge. The student participates through multiple choice type items.

Exercise 3A: For a given 3 x 3 system, the student constructs the C-matrix, computes the column sums, and decides whether or not the iteration will converge.

4. Row Sum Criteria as Sufficient Conditions for Convergence. The row sum criteria $\sum_j |c_{ij}| < 1$ for $i=1,\ldots,n$ is described.

Exercise 4A: For a given 4 x 4 system, the student constructs the C-matrix and computes the row sums.

The student is told if either the row sum or column sum conditions are satisfied, then the method of simultaneous displacements will converge.

5. Dominance Tests for Convergence and Review of Lesson. Describe the row sum and column sum tests in terms of the diagonal dominance of A for the system $Ax=b$.

Exercise 5A: The student tests diagonal dominance of a given 4 x 4 system.

Lesson 16: Numerical Differentiation

Purpose

Introduce the concept of the order of an approximation and the notation $O(h^k)$. Derive $O(h)$ and $O(h^2)$-approximations to $f'(x_1)$. Derive an $O(h^2)$-approximation to $f''(x_1)$. Demonstrate the computational difficulties of these approximations in the presence of round-off error.

Prerequisites

The lesson depends only on knowledge of Lessons A and B.

Lesson Outline

1. Order of an Approximation. Define an approximation $D(h)$ to some number A to be of order h if $\lim_{h\to 0}((A-D(h))/h)$ is a constant.

Exercise 1A: Given the Taylor formula $f(x_0+h)=f(x_0)+hf'(x_0)+h^2f''(z)/2$ where $x_0<z<x_0+h$, determine the order of $D(h)=(f(x_0+h)-f(x_0))/h$ if $A=f'(x_0)$.

Introduce the notation $D(h)=O(h^k)$ as meaning the approximation $D(h)$ is of order h^k.

Exercise 1B: Consider the two Taylor formulas

$$f(x_0+h)=f(x_0)+hf'(x_0)+h^2f''(x_0)/2+h^3f'''(z_1)/6 \text{ and}$$
$$f(x_0-h)=f(x_0)-hf'(x_0)+h^2f''(x_0)/2-h^3f'''(z_2)/6$$

where $x_0< z_1< x_0+h$ and $x_0-h<z_2<x_0$. Subtract the second formula from the first and let $D(h)=(f(x_0+h)-f(x_0-h))/2h$. Find an expression for $f'(x_0)-D(h)$.

Exercise 1C: In Exercise 1B, $f'(x_0)-D(h)=-h^2(f'''(z_1)+f'''(z_2))/12$ and, assuming continuity of f''', $\lim_{h\to 0}((f'(x_0)-D(h))/h^2)=-f'''(x_0)/6$. What is the order of $D(h)$?

2. Functions Tabulated on an Equally Spaced Set of Points. Define equally spaced points with spacing h.

Exercise 2A: Given function values on an equally spaced set of points $x_0,\ldots,x_5$, the student computes the values of $D(h)=(f(x_i+h)-f(x_i))/h$ and $D(h)=(f(x_i+h)-f(x_i-h))/2h$ for values of i.

Discuss the need for more knowledge about the function f prior to establishing error bounds on $D(h)-f'(x_i)$.

3. Numerical Approximation of $f'(x)$. Define the $O(h)$-approximation to $f'(x_i)$ as $D(h)=(f(x_{i+1})-f(x_i))/h$. Show that $|e(h)|=|f'(x_i)-D(h)|<(h/2)\max|f''(x)|$ for $x\epsilon[x_o,x_n]$. The student participates through multiple choice items.

Exercise 3A: A table of values for $f(x)=x^3-2x$ is presented for $x_i=-.1,0,.1,.2,.3$. On $[x_0,x_4]$, what is $\max|e(h)|$? At x_o, what is $D(h)$?

Define the $O(h^2)$-approximation to $f'(x_i)$ as $D(h)=(f(x_{i+1})-f(x_{i-1}))/(2h)$. Find an expression for $e(h)=f'(x_i)-D(h)$ from Exercise 1B. The student participates through multiple choice items.

Exercise 3B: Using the table in Exercise 3A, find a value of x_i for which the $O(h^2)$-approximation to $D(h)$ cannot be applied. Using $f(x)=x^3-2x$, compute $\max|e(h)|$. Compute the actual errors at $x_1=0$ and $x_3=.2$ and observe that they are less than $\max|e(h)|$.

Construct a table of $f'(x_i)$, $D(h)=(f(x_{i+1})-f(x_i))/h$, and $D(h)=(f(x_{i+1})-f(x_{i-1}))/(2h)$ for $x_1=0$, $x_2=.1$, and $x_3=.2$ and have the student observe the accuracy.

Derive the $O(h^2)$-approximations to $f'(x_o)$ and $f'(x_n)$, as $D(h)=(-3f(x_o)+4f(x_1)-f(x_2))/(2h)$ where $f'(x_o)-D(h)=h^2f'''(z_o)/3$ and $D(h)=(3f(x_n)-4f(x_{n-1})+f(x_{n-2}))/(2h)$ where $f'(x_n)-D(h)=h^2f'''(z_n)/3$. The student participates through multiple choice type items.

Exercise 3C: Using a set of tabulated values for the function $f(x)=x^2+2x$, the student computes $D(h)$ for $f'(x_o)$, $\max|f'(x_o)-D(h)|$, and $D(h)$ for x_n.

Define the $O(h^2)$-approximations of $f''(x_i)$ for $1< i< n$ as $D2(h)=(f(x_{i+1})-2f(x_i)+f(x_{i-1}))/h^2$ where $f''(x_i)-D2(h)=-(h^2)f''''(z_i)/12$.

The student participates through multiple choice type items.

Exercise 3D: Using a set of tabulated values for $f(x)=x^5+2x$, compute D2(h) for x_3. Find $\max | f''(x_3)-D(h) |$. For what values of x_i will D2(h) not apply?

5. Computational Accuracy of Numerical Differentiation. Introduce once again the two operators $D(h)=(f(x_{i+1})-f(x_{i-1}))/(2h)$ and $D2(h)=(f(x_{i+1})-2f(x_i)+f(x_{i-1}))/h^2$ with their respective error formulas.

Exercise 5A: Assuming continuity of $f'''(x)$, what is $\lim(f'(x_i)-D(h))$? Assuming continuity of $f''''(x)$, what is $\lim(f''(x_i)-D2(h))$?

Discuss the possible effects of round-off error on the limits in Exercise 5A.

Exercise 5B: For $f(x)=e^x$, the student calculates $f'(0)$ and $f''(0)$. The student then specifies various values of h, letting h tend to 0, and the values of $D(h)=(e^h-e^{-h})/(2h)$, $D2(h)=(e^h-2+e^{-h})/h^2$ are printed. The student observes that round-off error eventually dominates the error and notes a local optimum accuracy for D(h) around $h=10^{-5}$ and a local optimum accuracy for D2(h) around $h=.5\cdot 10^{-3}$.

Lesson 17: Extrapolation to the Limit

Purpose

Introduce the concept of extrapolation to the limit for differentiation by elimination of lower order terms in the expression for the error. By numerical examples, demonstrate the power of this technique up to the point where round-off dominates the error.

Prerequisites

Total familiarity with the concepts in Lesson 16 is assumed.

Lesson Outline

1. Review of the Order of an Approximation. Restate the definition of $O(h^k)$.

Exercise 1A: The student determines the order of the basic differentiation operator to be used in this lesson, $D(h)=(f(x_{i+1})-f(x_{i-1}))/(2h)$.

Derive the following properties of order h^k operators, $A(h)$ and $B(h)$: $A(h)+B(h)=O(h^k)$ and $M\cdot A(h)=O(h^k)$ where M is a constant. The student participates through constructed responses.

Exercise 1B: Suppose $\lim(z_i)=\lim(w_i)=x_0$ as $h\rightarrow 0$ and $f(x)$ and all derivatives in question are continuous. Furthermore, suppose $A(h)=h^2f'''(z_1)/2$ and $B(h)=h^2f'''(w_1)/6$. Compute the order of the following approximations: $A(h)+B(h)$, $h\cdot B(h)$, $A(h)\cdot B(h)$, $A(h/2)$, and $A(h^4)/B(h)$ where $f'''(x_0)\neq 0$.

2. Simple Extrapolation for Differentiation. From the Taylor formulas of $f(x_0+h)$ and $f(x_0-h)$ expanded about x_0, express $D(h)=(f(x_0+h)-f(x_0-h))/(2h)=a_0f(x_0)+\ldots+a_4f''''(x_0)+a_5(f'''''(z_0)+f'''''(z_1))$.

Exercise 2A: Compute the values of the coefficients $a_0,\ldots,a_5$ in terms of h. The resulting formula can be written as $D(h)=f'(x_0)+h^2f'''(x_0)+O(\text{what?})$. What is the order of $D(h)$? Replacing h by h/2, $D(h/2)=f'(x_0)+a_2h^2/4+O(\text{what?})$.

Using the formulas $D(h)=f'(x_0)+a_2h^2+O(h^4)$ and $D(h/2)=f'(x_0)+a_2h^2/4+O(h^4)$, form $D1(h/2)=(4D(h/2)-D(h))/3$.

Exercise 2B: What is the order of $D1(h/2)$?

Review the significance of obtaining the higher order approximation $D1(h/2)$.

Exercise 2C: The student is given a set of tabulated points for $f(x)=x^5$ from $x_0=.3$ to $x_4=.7$ with spacing .1. At the point $x_2=.5$, use $h=.2$ to compute the values $D(h)$, $D(h/2)$, and $D1(h/2)$. Compare these numbers with $f'(.5)=.3125$.

3. Repeated Extrapolation for Differentiation. Assuming $f(x)$ has continuous derivatives through the seventh order, write the expansions

$$f(x_0+h)=f(x_0)+hf'(x_0)+\ldots+h^7f^{(7)}(z_0)/5040$$

$$f(x_0-h)=f(x_0)-hf'(x_0)+\ldots-h^7f^{(7)}(z_1)/5040$$

and express

$$D(h)=(f(x_0+h)-f(x_0-h))/(2h)=a_0f(x_0)+\ldots+a_6f^{(6)}(x_0)+a_7(f^{(7)}(z_0)+f^{(7)}(z_1)).$$

Exercise 3A: Compute a_5, a_6, and a_7 in terms of h. Letting $b_1=f'''(x_0)/6$ and $b_2=f^{(5)}(x_0)/120$,

$$D(h)=f'(x_0)+b_1h^2+b_2h^4+O(\text{what?}),$$

$$D(h/2)=f'(x_0)+b_1h^2/4+b_2h^4/16+O(\text{what?}), \text{ and}$$

$$D(h/4)=f'(x_0)+b_1h^2/16+b_2h^4/256+O(\text{what?}).$$

Compute c_1, c_2, and c_3 for each of the simple extrapolations

$$D1(h/2)=(4D(h/2)-D(h))/3=f'(x_0)+c_1h^2+c_2h^4+O(c_3) \text{ and}$$

$$D1(h/4)=(4D(h/4)-D(h/2))/3=f'(x_0)+c_1h^2+c_2h^4+O(c_3).$$

Exercise 3B: Compute the value of M so that $D2(h/4)=(M\cdot D1(h/4)-D1(h/2))/(M-1)$ is an $O(h^6)$ approximation to $f'(x_0)$.

Exercise 3C: The student is given tabulated values of $f(x)=x^7$ for $x_0=.1$, $x_i=.1(i+2), i=1,\ldots,5$, and $x_6=.9$. For $h=.4$ and $x_3=.5$, compute the approximations $D(h)$, $D(h/2)$, $D(h/4)$, $D1(h/2)$, $D1(h/4)$, and $D2(h/4)$ and compare these values with the true solution $f'(.5)=.109375$.

4. Extrapolation to the Limit for Differentiation. Specify the general formulas and the construction of a triangular table for extrapolation to the limit based on n tabulated values of $f(x)$.

Exercise 4A: The student is given $f(x_i)=23.4609375/x_i$ for $x_4=.3$ and $x_i=.3+\text{sgn}(i-4)2^{|i-4|}h/16$, $i=0,\ldots,8, i\neq 4$. Using $h=.2$, the student computes the approximations $D(h)$, $D(h/2)$, $D(h/4)$, $D(h/8)$, $D1(h/2)$, $D1(h/4)$, $D1(h/8)$, $D2(h/4)$, $D2(h/8)$, and $D3(h/8)$ and compares these values with the true solution $f'(.3)=-260.670833\ldots$.

Explain the procedure for estimating the number of correct digits in an approximation by comparing successive diagonal entries in the extrapolation table. Point out the dangers of trying to extrapolate beyond the bounds of round-off error.

Lesson 18: Numerical Integration--The Trapezoidal Rule

Purpose

Introduce the notation for the integral sign. Develop the necessary background theory in order to develop the error formula for the trapezoidal rule. Define the trapezoidal rule and describe its geometric significance. Formally derive an expression for the error formula and demonstrate how maximum bounds might be placed on the error.

Prerequisites

This lesson is not dependent on the concepts in Lessons 1-17. However, a study of Lessons 1-17 will add to the maturity of the student in the area of numerical approximations and contribute to the overall performance. If the student is to progress to Lesson 19, then the concepts in Lessons 16-18 will be needed.

Lesson Outline

1. Notation for the Integral. Define the teletype notation for Integral(f(x);[A,B]) to denote $\int_A^B f(x)dx$. The student becomes familiar with the notation by finding the definite integral of several functions.

2. Second Theorem of the Mean for Integrals. State the theorem: If $f(x)$ and $g(x)$ are continuous on the interval $[A,B]$ and if $g(x)$ does not change sign on $[A,B]$, then there is a number $A<z<B$ so

$$\int_A^B f(x)g(x)dx=f(z)\int_A^B g(x)dx.$$

Exercise 2A: Suppose $f(x)=LN(x),g(x)=1/x$, and we wish to find $\int_A^B (LN(x)/x)dx$ where $[A,B]=[1,e]$. Does $g(x)$ change sign on $[A,B]$? Apply the above theorem to express the integral in terms of z. Find the exact value of the integral. Specify the value $1<z<e$ which yields the equality of the second theorem of the mean.

Point out to the student that the exact value of z is usually not known and that the future development will assume only its existence in (A,B).

Formally prove the second theorem of the mean. The student participates through multiple choice items.

Exercise 2B: Let $f(x)=\cos(x)$, $g(x)=\sin(x)$, $A=0$, and $B=PI/2$. Does the second theorem of the mean guarantee the existence of $A<z<B$?

Exercise 2C: The student repeats Exercise 2B with the roles of $f(x)$ and $g(x)$ reversed and $[A,B]=[0,PI]$.

Exercise 2D: Let $h_1(x)=x^3$, $h_2(x)=e^{-x}$, $A=-1$, and $B=1$. Apply the mean value theorem to express $\int_A^B h_1(x)h_2(x)dx$ in terms of z for some $A<z<B$.

3. Review of Rolle's Theorem. State Rolle's theorem:

Let $f(x)$ be continuous on $[A,B]$ and differentiable on (A,B) and suppose $f(A)=f(B)=0$. Then there is a point $A<z<B$ so that $f'(z)=0$.

Exercise 3A: Let $f(x)=x^2-1$ and $[A,B]=[-1,1]$. Does Rolle's theorem apply? Name a point z in $[-1,1]$ so that $f'(z)=0$.

Exercise 3B: Let $f(x)=\sin(x)$ and $[A,B]=[0,PI]$. Apply Rolle's theorem to find the value of z.

Point out to the student that the exact value of z is usually not known, but we will depend on Rolle's theorem for its existence.

4. Error in Linear Approximation to a Function. Geometrically describe the process of approximating a function $f(x)$ on an interval $[x_0,x_1]$ by the straight line $p(x)=f(x_0)+(f(x_1)-f(x_0))(x-x_0)/h$ where $x_1-x_0=h$. Describe the objective as deriving some expression for $e(x)=f(x)-p(x)$. Introduce the auxiliary function in the variable s with fixed x by

$$g(s)=f(s)-p(s)-(s-x_0)(s-x_1)(f(x)-p(x))/((x-x_0)(x-x_1)).$$

Exercise 4A: Compute $g(x_0)$, $g(x_1)$, and $g(x)$. $g(s)$ has at least how many zeros in $[x_0, x_1]$? Assuming that $f(s)$, $f'(s)$, and $f''(s)$ are continuous, $g'(s)$ has at least how many zeros in $[x_0,x_1]$? $g''(s)$ has at least how many zeros in $[x_0,x_1]$?

Exercise 4B: Denote the zero of $g''(s)$ by z, i.e. $g''(z)=0$ where $x_0<z<x_1$. To form $f(x)-p(x)$, compute $g''(s)$ and evaluate at z. What is $p''(s)$ and the second derivative with respect to s of $(s-x_0)(s-x_1)$? Using $g''(z)=0$, the student observes $0=f''(z)-2(f(x)-p(x))/((x-x_0)(x-x_1))$ and $e(x)=f(x)-p(x)=f''(z)(x-x_0)(x-x_1)/2$.

5. Derivation of the Trapezoidal Rule and Its Error. Define the trapezoidal rule on the interval $[x_0,x_1]$ as $\int_{x_0}^{x_1}p(x)dx$ where $p(x)$ is the straight line approximation to $f(x)$ given in Section 4.

Exercise 5A: Write an expression for $\int_{x_0}^{x_1}p(x)dx$ using $x_1-x_0=h$ to eliminate x_0 and x_1 from your answer.

The student is told to note that $(h/2)(f(x_0)+f(x_1))$ is the area of a trapezoid and hence the name trapezoidal rule. The student is told to review the error formula $p(x)-f(x)$ in Section 4 and the second theorem of the mean in Section 2 prior to discussing a possible error formula for

$$e(h)=\int_{x_0}^{x_1}(f(x)-p(x))dx.$$

The error formula $e(h)=-h^3f''(z)/12$ where $x_0<z<x_1$ is derived. The student participates through constructed responses and multiple choice items.

Exercise 5B: Let $f(x)=e^{-x}$, $x_0=-1$, and $x_1=0$. Compute $\int_{x_0}^{x_1}f(x)dx$, $\int_{x_0}^{x_1}p(x)dx$, $e(h)=\int_{x_0}^{x_1}(f(x)-p(x))\,dx$ in terms of z using the error formula, and compute max $|e(h)|$ on $[x_0,x_1]$.

6. General Application of the Trapezoidal Rule. Discuss the process of dividing an interval [A,B] into n subdivisions of length $x_{i+1}-x_i=h$ and summing the trapezoidal rules over all intervals to approximate $\int_A^B f(x)dx$. The student is required to write the form of the trapezoidal rule on the interval $[x_i,x_{i+1}]$.

Exercise 6A: Suppose $f(x)=\sin(x)$ and we wish to approximate $\int_0^{1.5} f(x)dx$ by the trapezoidal rule. If $h=.5$, compute the number of subdivisions n, x_0, x_1, x_2, x_3, and the trapezoidal approximation over [A,B].

Describe the process of finding the error for the trapezoidal rule, summing over the n intervals, to arrive at $e(h)=-h^2(b-a)f''(z)/12$.

Exercise 6B: Write the error in terms of h and z for the approximation to $f(x)$ in Exercise 6A. Choose a spacing h so that $\max|e(h)| \leq .5\cdot 10^{-4}$.

Point out to the student that finding a value of h by bounding $\max|e(h)|$ gives an upper bound on the needed number of subdivisions.

Lesson 19: Romberg Integration

Purpose

Develop the error formula of the trapezoidal rule as the error expression $a_1h^2+a_2h^4+\ldots+O(h^{2k})$ depending on the continuity of the function being integrated. Demonstrate the method of extrapolation to the limit to increase the accuracy of the trapezoidal rule.

Prerequisites

The student should be familiar with the basic differentiation formulas from Lesson 16, extrapolation to the limit from Lesson 17, and the trapezoidal formula from Lesson 18.

Lesson Outline

1. Introduction. Restate the trapezoidal rule and the associated error formula. State that the first purpose of the lesson is to derive the trapezoidal approximations $T_0(h)=\int_A^B f(x)dx+ah^2+O(h^4)$ so that $T_0(h/2)=\int_A^B f(x)dx+ah^2/4+O(h^4)$. Extrapolation $T_1(h/2)=(M\cdot T_0(h/2)-T_0(h))/(M-1)$ will then give an improved result. The student is asked to determine the needed value of M.

2. Basic Differentiation Formulas. Derive the numerical differentiation formulas

a. $f'(x_0)=(f(x_1)-f(x_0))/h-hf''(x_0)/2-h^2f'''(x_0)/6+0(h^3)$

b. $f''(x_0)=(f(x_2)-2f(x_1)+f(x_0))/h^2-hf'''(x_0)+0(h^2)$

c. $f'''(x_0)=(f(x_3)-3f(x_2)+3f(x_1)-f(x_0))/h^3+0(h)$

by Taylor series expansions for use in the later derivation of the trapezoidal error formula. The student participates through constructed responses to determine coefficients of the formulas and also through some multiple choice items.

3. General Formulation of the Trapezoidal Rule. State the Taylor formula

$$f(x)=f(x_0)+(x-x_0)f'(x_0)+(x-x_0)^2f''(x_0)/2+(x-x_0)^3f'''(x_0)/6+(x-x_0)^4f''''(z)/24.$$

Exercise 3A: Integration of both sides of the above formula yields

$$\int_{x_0}^{x_1}f(x)dx=af(x_0)+bf'(x_0)+cf''(x_0)+df'''(x_0)+ef''''(w).$$

The student writes the expressions for a, b, c, d, and e in terms of h.

The result of Exercise 3A is

$$\int_{x_0}^{x_1}f(x)=hf(x_0)+h^2f'(x_0)/2+h^3f''(x_0)/6+h^4f'''(x_0)/24+0(h^5).$$

Replace $f'(x_0)$, $f''(x_0)$, and $f'''(x_0)$ in the last formula by differentiation formulas a, b, and c to obtain

$$\int_{x_0}^{x_1}f(x)dx=h(f(x_0)+f(x_1))/2-h(f(x_2)-2f(x_1)+f(x_0))/12$$
$$+h(f(x_3)-3f(x_2)+3f(x_1)-f(x_0))/24+0(h^5).$$

The student participates in the replacement process through multiple choice items.

State the general result on $[x_i,x_{i+1}]$ as

$$\int_{x_i}^{x_{i+1}} f(x)dx=h(f(x_i)+f(x_{i+1}))/2-h(f(x_{i+2})-2f(x_{i+1})+f(x_i))/12$$

$$+h(f(x_{i+3})-3f(x_{i+2})+3f(x_{i+1})-f(x_i))+O(h^5).$$

Summing over n subdivision of [A,B], state the result as

$$\int_A^B f(x)dx=(h/2)\sum_{i=0}^{n-1}(f(x_i)+f(x_{i+1}))-(h/12)\sum_{i=0}^{n-1}(f(x_{i+2})-2f(x_{i+1})+f(x_i))$$

$$+(h/24)\sum_{i=0}^{n-1}(f(x_{i+3})-3f(x_{i+2})+3f(x_{i+1})-f(x_i))+\sum_{i=0}^{n-1}O(h^5).$$

Exercise 3A: $(h/2)\sum_{i=0}^{n-1}(f(x_i)+f(x_{i+1}))$ is the trapezoidal approximation. The student determines expressions for $\sum_{i=0}^{n-1}(f(x_{i+2})-2f(x_{i+1})+f(x_i))$ using only four terms and $\sum_{i=0}^{n-1}(f(x_{i+3})-3f(x_{i+2})+3f(x_{i+1})-f(x_i))$ using only six terms. $\sum_{i=0}^{n-1}O(h^5)=O(\text{what?})$

Exercise 3A yields the expression

$$\int_A^B f(x)dx=T_0(h)-h(f(x_0)-f(x_1)+f(x_{n+1})-f(x_n))/12+h(f(x_{n+2})-2f(x_{n+1})$$

$$+f(x_n)-f(x_2)+2f(x_1)-f(x_0))/24+O(h^4).$$

Use the differentiation formulas a and b from Section 2 to establish

$$T_0(h)=\int_A^B f(x)dx-h^2(f'(x_n)-f'(x_0))/12+O(h^4).$$

The student participates through multiple choice type items.

4. Romberg Integration--Simple Extrapolation. Review the final result of Section 3, namely, $T_0(h)=I+ch^2+O(h^4)$ and $T_0(h/2)=I+ch^2/4+O(h^4)$ where c is a constant and $I=\int_A^B f(x)dx$. Ask the student to write an $O(h^4)$-approximation to I in terms of $T_0(h)$ and $T_0(h/2)$.

Exercise 4A: Suppose we wish to approximate $\int_1^2 (1/x)dx$. Using $h=.5$, write the expressions for $T_0(h)$, $T_0(h/2)$, and the extrapolated result $T_1(h/2)$.

5. Romberg Integration--Repeated Extrapolation. State without derivation that a more general result can be obtained with more time and effort, namely,

a. $T_0(h)=I+c_1h^2+c_2h^4+O(h^6)$.

Exercise: Replace h by h/2 to obtain $T_0(h/2)=I+d_1h^2+d_2h^4+O(h^6)$. Determine the values for d_1 and d_2 in terms of h, c_1, and c_2. Replace h by h/4 in formula a to obtain $T_0(h/4)=I+e_1h^2+e_2h^4+O(h^6)$. Determine the values for e_1 and e_2. We now have the additional expressions:

b. $T_0(h/2)=I+c_1h^2/4+c_2h^4/16+O(h^6)$ and

c. $T_0(h/4)=I+c_1h^2/16+c_2h^4/256+O(h^6)$.

Use simple extrapolation on a and b to obtain $T_1(h/2)$ as an $O(h^4)$-approximation to I. Write the formula. Use simple extrapolation on b and c to obtain $T_1(h/4)$ as another $O(h^4)$-approximation to I. Write the formula. We now have the $O(h^4)$-approximations:

d. $T_1(h/2)=I-3c_2h^4/4+O(h^6)$ and

e. $T_1(h/4)=I-3c_2h^4/64+O(h^6)$.

Use simple extrapolation on d and e to obtain an $O(h^6)$-approximation to I and write the formula.

6. Romberg Integration--Extrapolation to the Limit. Explain the general procedure for extrapolation to the limit by displaying a general extrapolation table and the method of computing entries in the table. There is no student participation in this section.

Lesson 20: Numerical Integration--Simpson's Rule

Purpose

Derive Simpson's formula over an interval of length 2h by simple extrapolation using the trapezoidal rule. Introduce the error formula for Simpson's rule for an interval of length 2h. Derive the general form of Simpson's rule and the associated error formula over n intervals of length 2h where $n=(B-A)/(2h)$. Illustrate how the general error can be bounded to determine an upper bound on n (or lower bound on h) for a specified accuracy.

Prerequisites

The student must be familiar with the trapezoidal rule (see Lesson 18) and simple extrapolation (see Lesson 19).

Lesson Outline

1. Review of the Trapezoidal Rule. State the general formula and associated error expression based on n subdivisions of the interval $[A,B]$.

Exercise 1A: Write the trapezoidal approximation to $\int_1^3 \cos(x)dx$ using n=3 subdivisions. Write the error formula.

2. Review of Romberg Integration--Simple Extrapolation. State the principle of simple extrapolation to obtain an $O(h^4)$-approximation.

3. Simpson's Rule on an Interval of Length 2h. Consider f(x) on the interval $[x_0,x_2]$ where x_1 is the midpoint and $h=x_2-x_1=x_1-x_0$. The student is asked to write the trapezoidal approximation of $\int_{x_0}^{x_2} f(x)dx$ using n=1 subdivision and the trapezoidal rule for n=2 subdivisions. The student is asked to observe that the two applications of the trapezoidal rule gives $T_0(2h)=h(f(x_0)+f(x_2))=I+4ah^2+O(h^4)$ and $T_0(h)=(h/2)(f(x_0)+2f(x_1)+f(x_2))=I+ah^2+O(h^4)$. The student is asked to write the simple extrapolation $T_1(h)=(4T_0(h)-T_0(2h))/3$ in terms of $f(x_1)$ and h. The student is told that the resulting formula $S_0=(h/3)(f(x_0)+4f(x_1)+f(x_2))$ is known as Simpson's rule on an interval of length 2h.

4. General Form of Simpson's Rule. Simpson's rule on $[x_0,x_2]$ is stated as $S_0=(h/3)(f(x_0)+4f(x_1)+f(x_2))$ and on $[x_2,x_4]$ as $S_1=(h/3)(f(x_2)+4f(x_3)+f(x_4))$. The student forms the expression S_0+S_1 and thus observes the form of Simpson's rule on $[x_0,x_4]$ with four subdivisions. The formula is generalized for the student for 2n subdivisions.

Exercise 4A: How many evaluations of $f(x)$ are required in Simpson's rule $S_n=(h/3)(f(x_0)+4f(x_1)+2f(x_2)+\ldots+2f(x_{2n-2})+4f(x_{2n-1})+f(x_{2n}))$? How many subdivisions of length h are needed? The student observes the even number of subdivisions and odd number of points.

Exercise 4B: Suppose we wish to approximate $\int_0^3 \sin(x)dx$ by Simpson's rule using four subdivisions. n=? h=? The values of $(x_i,f(x_i))$ for i=0,...,2n are printed for the student. S_4=?

5. Error Formula in Simpson's Rule. The error formula $e_i(h)=-h^5f''''(z_i)/90$ for $x_{2i}<z_i<x_{2i+2}$ is given to the student without derivation as the accepted error in approximating $\int_{x_{2i}}^{x_{2i+2}} f(x)dx$ by Simpson's rule with two subdivisions. The errors are summed over the n double length intervals to obtain $e(h)=S_n-\int_{x_0}^{x_{2n}} f(x)dx=-h^4(x_{2n}-x_0)f''''(z)/180$ where $x_0<z<x_{2n}$. The student participates through multiple choice type items.

Exercise 5A: Suppose we wish to estimate $\int_1^2 \ln(x)dx$ by Simpson's rule and we wish to choose h so that $\max|e(h)|<10^{-4}/30$. Compute $f''''(x)$ and $\max|f''''(x)|$ on $[1,2]$. Choose h so that $h^4\max|f''''(x)|(b-a)/180<10^{-4}/30$. Thus choosing $h<.1$ means $n>$?. The student may choose various values of n and the computer will print $(x_i,f(x_i))$ for i=0,...,2n and S-Simpson's approximation.

Lesson 21: Numerical Integration of Ordinary Differential Equations by Taylor Series Approximations

Purpose

Describe the Taylor algorithms of orders 1, 2, and 3 for numerically approximating the solution to $y'=f(x,y)$ given an initial value (x_0,y_0). Demonstrate the deficiency of lower order methods and possible complexity of higher order methods. In this lesson, the order of a particular method is not rigorously established.

Prerequisites

The concepts from previous lessons are not needed here although a study of numerical differentiation and approximation of definite integrals (Lessons 16-20) serve as a good background.

Lesson Outline

1. Statement of the Initial Value Problem. Describe the problem as one of approximating the numerical values of $y(x)$ on an interval $[A,B]$, given the differential equation $y'=f(x,y)$ with initial known conditions $(A,y(A))$.

Exercise 1A: Suppose $y'=-e^{-x}+1$ with the known condition $y(1)=e^{-1}$. Then $y(x)=\int y'dx+c$ where c is the constant of integration. Compute $\int y'dx$ and use the initial condition to determine the value of c.

Point out to the student that not all functions can be explicitly integrated and thus we need approximation techniques.

2. Taylor's Algorithm of Order 1--Euler's Method. State the method as forming the Taylor formula $y(x+h)=y(x)+hy'(x)+h^2y''(z)/2$ where

$x<z<x+h$ and where all derivatives are assumed continuous. Euler's method comes by dropping the $O(h^2)$-term and stepping across the interval $[A,B]$ by $y_{i+1}=y_i+hy_i'=y_i+hf(x_i,y_i)$.

Exercise 2A: Suppose $y'=f(x,y)=-e^{-x}+1$ with $y(1)=e^{-1}$. Write the Taylor formula in terms of x, y, and z. Write the approximation formula in terms of x_i and y_i. Starting at $x_o=1$ with step-size $h=.25$, write y_1 as the approximation to $y(1.25)$. $y_2=?$ $y_3=?$ $y_4=?$ The student compares the approximate values with the true solution computed to fifteen decimal figures.

Exercise 2B: Let $f(x,y)=x\cdot\sin(y)$ with $x_o=3$ and $y_o=1.5$. Write Euler's method in terms of h, x_i, and y_i. Using $h=.1$, how many applications of Euler's method is needed to approximate $y(6)$?

3. Review of Notation for Partial and Total Derivatives. Define the notation to be used by the terminal for partial derivatives and give the definition of the total derivative of $f(x,y)$ with respect to x as $f'(x,y)=f_x+f_yf$.

Exercise 3A: Let $f(x,y)=x+y+x^3y^2$. Compute f_x, f_y, f_{xx}, f_{yy}, f_{xy}, and $f'(x,y)$.

Exercise 3B: Suppose we have the differential equation $y'=f(x,y)=\sin(x^2+y)$ where the solution $y(x)$ is a function of x. Compute $y''(x)$ in terms of x and y.

4. Taylor's Algorithm of Order 2. Derive the computational procedure $y_{i+1}=y_i+hf(x_i,y_i)+h^2(f_x(x_i,y_i)+f_y(x_i,y_i)f(x_i,y_i))/2$. The student participates through multiple choice type items.

Exercise 4A: Suppose $y'=y(1-x)/x$ and $y(1)=e^{-1}$. Compute f_x, f_y, and $f'(x,y)$ in terms of x and y. Write y_{i+1} in terms of x_i and y_i. The student then specifies various values of n as the total number of steps from $x_0=1$ to $x_n=3$. The computer responds with values $(x_i, y_i, y(x_i))$ for $i=0,\ldots,n$.

5. Taylor's Algorithm of Order 3. Derive the expression $f''(x,y)=f_{xx}+2f_{xy}f+f_y f_x+f_{yy}f^2+ff_y^2$ and the third order Taylor approximation.

$$y_{i+1}=y_i+hf(x_i,y_i)+h^2f'(x_i,y_i)/2+h^3f''(x_i,y_i)/6.$$

The student participates through multiple choice type items.

Exercise 5A: Suppose $y'=xy$ and $y(0)=1$. What is $f(x,y)$, $f'(x,y)$, and $f''(x,y)$? To apply the third order algorithm on the interval $[0,1]$, the student specifies various values of n. The computer responds with the numerical values of $(x_i, y_i, y(x_i))$ for $i=0,\ldots,n$.

6. Taylor's Algorithm of Order k. The general algorithm derived from truncation of a Taylor series after the kth derivative is described for the student. It is pointed out that for $k=1$ (Euler's Method), an extremely small step-size h is usually needed for reasonable accuracy, thus requiring a great deal of computational work. On the other hand, for large values of k, the higher order derivatives may be algebraically cumbersome. For this reason, Taylor's algorithm of order $k=2$ or 3 is popular. There is no student participation in this section.

Lesson 22: Second Order Runge-Kutta Methods

Purpose

Derive the class of second order Runge-Kutta methods as an alternative to the Taylor algorithm of order 2. Demonstrate that, at the cost of two evaluations of the function, no evaluations of derivatives are needed. Demonstrate the numerical accuracy of this class for the Improved Euler's method and the Modified Euler's method. Derive the principal error function for the Improved Euler's method as the major contributing factor and explain how its magnitude may be difficult to estimate.

Prerequisites

The student is expected to know Taylor's algorithm of order 2 from Lesson 21.

Lesson Outline

1. Introduction. State that the purpose of the lesson is to derive a class of methods which are equivalent in order to the Taylor algorithm of order 2 but need no evaluation of derivatives. Write the general formulas for second order Runge-Kutta methods. There is no student participation in this section.

2. Numerical Example of a Second Order Runge-Kutta Method. Define the special case

$$y_{i+1}=y_i+.5(K_1+K_2)$$
$$K_1=hf(x_i,y_i)$$
$$K_2=hf(x_i+h,y_i+K_1)$$

Exercise 2A: Suppose $y'=xy$. Compute K_1, K_2, and the computational formula y_{i+1}. Using $x_0=0$, $y_0=1$, and $h=.1$, compute y_1, x_1, y_2, and x_2. The exact values of the true solution $y(x_i)=e^{x_i^2/2}$ are printed and the student compares the values.

3. Optimal Parameters a, b, c, and d. State the overall procedure as comparing $y_{i+1}=y_i+aK_1+bK_2$, where $K_1=hf(x_i,y_i)$ and $K_2=hf(x_i+ch,y_i+dK_1)$,

with a Taylor expansion of $y(x_{i+1})$ to determine the best choice for a, b, c, and d. Give the student the general Taylor expansion

$$f(x+s,y+t)=f(x,y)+sf_x+tf_y+s^2f_{xx}/2+stf_{xy}+t^2f_{yy}/2+\ldots$$

Then

$$K_2/h=a_of+a_1f_x+a_2f_y+a_3f_{xx}+a_4f_{xy}+a_5f_{yy}+\ldots$$

with $s=ch$ and $t=dK_1$.

Exercise 3A: Determine $a_1,\ldots,a_5$ in terms of h, c, d, and K_1.

Ask the student to observe that substitution of $K_1=hf$ in the results of Exercise 3A yields.

$$K_2/h=f+hcf_x+dhff_y+h^2c^2f_{xx}/2+h^2cdf_{xy}f+h^2d^2f_{yy}f^2/2+O(h^3).$$

Exercise 3B: In the last expression for K_2/h, collect terms in powers of h to obtain $K_2/h=S_o+hS_1+h^2S_2+O(h^3)$. Write S_0, S_1, and S_2 in terms of c, d, f, f_x, f_y, f_{xx}, f_{yy}, and f_{xy}.

From the results of Exercise 3B, the student is asked to observe that $K_2=hf+h^2(cf_x+df_yf)+h^3(c^2f_{xx}/2+cdf_{xy}f+d^2f_{yy}f^2/2)+O(h^4)$
and $y_{i+1}=y_i+h(a+b)f+h^2b(cf_x+df_yf)+h^3b(c^2f_{xx}/2+cdf_{xy}f+d^2f_{yy}f^2/2)+O(h^4)$.
The student is then asked to compare the last formula with the standard Taylor expansion for functions of one variable
$y(x_{i+1})=y(x_i)+hf+h^2(f_x+f_yf)/2+h^3(f_{xx}+2ff_{xy}+f_{yy}f^2+f_xf_y+ff_y^2)/6+O(h^4)$.

Exercise 3C: Comparing the last two expressions, the best accuracy is obtained for a+b=? bc=bd=?

The student is asked to observe that the $O(h^3)$ terms cannot generally be equated and thus the local error is $O(h^3)$. Remarks are made about the total error over the interval $[x_o,x_n]$ being $O(h^2)$, but a rigorous discussion is not presented.

4. Special Cases and a Look at the Local Error. The student is asked to recall that the best values of a, b, c, and d satisfy $a+b=1$ and $bc=bd=\frac{1}{2}$.

Define the Improved Euler's method as that RK-method for which $a=b=\frac{1}{2}$ and $c=d=1$. Derive the local error formula

$$y_{i+1}-y(x_{i+1})=h^3(f_{xx}+2f_{xy}f+f_{yy}f^2-2f_xf_y-2ff_y^2)/12+O(h^4).$$

The student participates through constructed responses. The principal error term is defined as $g(x,y)$ where $y_{i+1}-y(x_{i+1})=h^3g(x_i,y_i)+O(h^4)$.

Exercise 4A: Suppose $y'=xy$. Find the principal error $g(x,y)$ for the Improved Euler's method.

Define the Modified Euler's method as the special case where $a=0$, $b=1$, and $c=d=\frac{1}{2}$.

Exercise 4B: Let $y'=y\cdot\sin(x)$, $x_0=PI$, and $y(PI)=e$. Applying the Modified Euler's method, calculate K_1 and K_2 in terms of h, x_i and y_i. Write the general computational formula $y_{i+1}=?$. The student may specify values of n and the computer will respond with the values $(x_i,y_i,y(x_i)=e^{-\cos(x_i)})$ for $i=0,\ldots,n$.

The general inability to accurately estimate the local error is discussed.

Lesson 23: Numerical Integration, Error Estimation, and Extrapolation

Purpose

Formally demonstrate that the total error $y_n-y(x_n)$ is $O(h^2)$ for the second order methods described in Lessons 22 and 23, thus warranting the name "second order". State a more general error formula and show how extrapolation improves the numerical result.

Prerequisites

The student is expected to be familiar with second order Runge-Kutta methods (Lesson 22), Taylor's algorithm of order 2 (Lesson 21), and simple extrapolation (Lessons 17 and 19).

Lesson Outline

1. Review of Second Order Methods for the Solution of $y'=f(x,y)$.

Display the computational formula for Taylor's algorithm of order 2.

Exercise 1A: Let $y'=ye^x$. Compute f_x, f_y, and the computational formula $y_{i+1}=?$.

Display the computational formulas for second order Runge-Kutta methods.

Exercise 1B: Using $a=b=\frac{1}{2}$ and $c=d=1$, write the expressions for K_1, K_2, and the Improved Euler's method $y_{i+1}=?$.

Exercise 1C: Using $a=0$, $b=1$, and $c=d=\frac{1}{2}$, write the expressions for K_1, K_2, and the Modified Euler's method $y_{i+1}=?$.

2. Estimation of the Cumulative Error $y_n-y(x_n)$. State the approximate solution as $y_{i+1}=y_i+hT(x_i,y_i)$ and define $T(x,y)$ for both Taylor's algorithm and Runge-Kutta methods. Using the exact solution $y(x_{i+1})=y(x_i)+hT(x_i,y(x_i))+O(h^3)$ and the error notation $e_i=y_i-y(x_i)$, establish that $|e_n|<(e^{(x_n-x_o)m}-1)O(h^2)=O(h^2)$ assuming that T and T_y are bounded and continuous with $|T_y|<m$. The student participates in a somewhat lengthy analysis through some constructed responses and a great number of multiple choice items.

3. **Practical Estimation of the Error $y_n - y(x_n)$.** Without proof, describe for the student the more general result $e_n(h) = y_n - y(x_n) = ch^2 + O(h^3)$ where c is a constant. Using a step-size h/2, $e_n(h/2) = ch^2/4 + O(h^3)$. The student is asked to construct an extrapolation formula which will improve the approximation to $y(x_n)$.

Exercise 3A: Let $y' = x + y$ and $y(0) = 1$. What is the computational formula for Taylor's algorithm of order 2? We wish to estimate y(1). The student chooses a value of $.05 < h < 1$ so that n is an integer. The computer responds with the values of n and $(x_i, y_i, y(x_i) = 2e^{x_i} - x_i - 1)$ for $i = 0, \ldots, n$. What is h/2? The computer responds with the new values of n and $(x_i, y_i, y(x_i))$ for $i = 0, 1, \ldots, n$. What is the extrapolated value? The student is asked to observe the agreement between $y_n(h/2)$ and the extrapolated value to obtain a lower bound on the number of correct digits.

APPENDIX C

QUESTIONNAIRE AND EXAMINATIONS

Post-Experiment Questionnaire for CAI Group

CIRCLE THE ANSWER THAT BEST DESCRIBES YOUR OPINION OR REACTION

1. I purposely typed an equivalent form of what I knew to be the correct algebraic expression just to see what would happen.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

2. The examples and exercises in the Tutorial Mode clarified the concepts and helped me gain additional insight into the theory.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

3. The Problem Mode should be eliminated in favor of programming the problems in the conventional Fortran manner.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

4. The Investigation Mode should be eliminated because I could have accomplished the same thing more quickly and more flexibly by conventional Fortran programming.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

5. In view of the fact that I had a textbook, the most useful mode(s) of instruction is (are)

: Tutorial

: Problem

: Investigation

: Uncertain

: Tutorial and Problem (both equal)

: Tutorial and Investigation (equal)

: Problem and Investigation (equal)

: Tutorial, Problem and Investigation (all three about equal)

6. If only two modes of instruction were possible, the one I would choose to drop is the

:	:	:	:
Lesson	Problem	Investigation	Uncertain

7. If only one mode of instruction were possible, the one I would choose to retain is the

:	:	:	:
Lesson	Problem	Investigation	Uncertain

8. I was more involved with pushing keys than concentrating on the material.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

9. I felt tense or ill at ease at the teletype.

:	:	:	:	:
Almost never	seldom	about half the time	usually	almost always

10. When typing mathematical expressions, I found myself concentrating on avoiding Fortran errors and forgot the question or material leading to the question.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

11. When the computer was typing information, I became impatient.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

12. The Fortran notation for mathematical expressions made the material harder to read.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

13. Automatic computation of arithmetic results by the computer helped me to concentrate more on the analysis of the theory, formulation of the problems, and the interpretation of the results.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

14. When I answered wrong, it was an attempt to "fool" the computer or just to see what would happen.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

15. I feel that more can be gained from the conventional classroom than from the Tutorial Mode.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

16. Teletype noise distracted my attention.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

17. I learned more from reading the textbook than I did from the Tutorial Mode

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

18. Compared with the previous course material, I found the material on numerical differentiation, numerical integration, and numerical solution of differential equations to be considerably more difficult.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

19. Compared with the previous course material, I had considerably more difficulty reading the linear teletype notation in lessons on differentiation, integration, and differential equations.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

20. The fact that some developments in the sections on numerical differentiation, numerical integration, and differential equations deviated considerably from developments in the textbook made the material harder to understand and learn.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

21. I felt that I lacked the proper prerequisite knowledge for studying the course material.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

22. The method by which I was told whether or not I had given a correct answer became monotonous.

:	:	:	:	:
strongly disagree	disagree	uncertain	agree	strongly agree

23. Whenever I typed HELP in the Tutorial Mode, I was given information which actually helped me to understand the concept and construct the right answer.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

24. Whenever I was given the correct answer, I was also given an adequate explanation of why it was correct and I could determine what was wrong with my answer.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

25. Whenever I typed HELP or answered incorrectly, it was because I was not inspired to think or I really didn't care.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

26. Whenever I typed HELP, I really knew the right answer and was only trying to gain additional information.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

27. The lesson material was on the average

:	:	:	:	:
too easy	easy	challenging	difficult	too difficult

28. The Tutorial Mode clarified the outside reading assignment and helped me to gain a deeper understanding of the course material.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

29. When solving problems in the Problem Mode, I usually needed

:	:	:
less help	more help	the help it now provides (i.e., told only when wrong)

30. The investigation Mode provided an outlet for solving my own problems and answering my own questions.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

31. In the Tutorial Mode, I understood the relevance between the question and the lesson material.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

32. The lesson material was too repetitious.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

33. The computer lesson seemed organized.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

34. I knew when I needed to type HELP.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

35. The notes produced by the computer lesson were acceptable for home study or review.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

36. I did not understand the material but was forced to go on without adequate explanation.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

37. I found myself trying to get through the material rather than learning it.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

38. I guessed at answers to questions when I didn't know the correct answer rather than type HELP.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

39. The computer lesson was boring and tiresome.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

40. A picture, graph, or diagram would have clarified the concepts and helped me to learn more rapidly.

:	:	:	:	:
almost never	seldom	about half the time	usually	almost always

41. At the start, my enthusiasm for studying numerical analysis by computer was

:	:	:	:	:
very low	low	normal	high	very high

42. At the present, my enthusiasm for studying numerical analysis by computer is

:	:	:	:	:
very low	low	normal	high	very high

43. I feel that my overall knowledge of computer-presented course material is

:	:	:	:
poor	fair	good	excellent

44. Compared to my actual knowledge and understanding of the course material, I feel that my average performance on examinations has been

:	:	:	:	:
very low	low	about right	high	very high

45. The average amount of time I spent in preparation prior to the Tutorial Mode was

:	:	:	:	:
less than 15 minutes	15-30 minutes	30-45 minutes	45-60 minutes	greater than 60 minutes

Table 20. Individual Responses to Questionnaire Items

Item No.	Student 34	Student 35	Student 36	Student 37
1	seldom	seldom	half the time	seldom
2	usually	usually	usually	usually
3	almost never	half the time	seldom	almost never
4	seldom	seldom	seldom	almost never
5	tutorial	tut. and prob.	prob. and inv.	tut. and prob.
6	investigation	investigation	uncertain	investigation
7	tutorial	tutorial	investigation	tutorial
8	seldom	seldom	seldom	almost never
9	almost never	almost never	seldom	almost never
10	half the time	seldom	seldom	seldom
11	almost never	seldom	seldom	seldom
12	almost never	seldom	usually	half the time
13	half the time	half the time	half the time	almost always
14	seldom	seldom	seldom	almost never
15	seldom	seldom	half the time	half the time
16	half the time	almost never	seldom	seldom
17	almost never	seldom	seldom	seldom
18	usually	half the time	usually	usually
19	seldom	seldom	usually	seldom
20	half the time	half the time	almost always	seldom
21	seldom	half the time	seldom	almost never
22	disagree	agree	disagree	disagree
23	usually	half the time	usually	usually
24	usually	usually	usually	usually
25	seldom	seldom	seldom	almost never
26	half the time	seldom	seldom	seldom
27	challenging	challenging	challenging	challenging
28	usually	half the time	seldom	usually
29	help it now provides	help it now provides	help it now provides	help it now provides
30	half the time	seldom	seldom	seldom
31	usually	half the time	usually	usually
32	almost never	seldom	seldom	seldom
33	almost always	usually	almost always	usually
34	usually	usually	almost always	half the time
35	half the time	half the time	usually	almost always
36	seldom	seldom	seldom	almost never
37	seldom	seldom	seldom	half the time
38	seldom	seldom	seldom	half the time
39	seldom	seldom	seldom	seldom
40	half the time	half the time	half the time	seldom
41	very high	high	very high	high
42	very high	normal	very high	high
43	good	fair	fair	good
44	about right	low	low	high
45	30-45 mins.	15-30 mins.	15-30 mins.	15-30 mins.

Examination 1

WORK ALL PROBLEMS. EACH PROBLEM IS WORTH 20 POINTS.

Problem 1: Describe the following concepts in mathematical terms. Define all symbols that you introduce.

(a) any three conditions under which you might expect the iteration $x_{k+1}=g(x_k)$ to diverge

(b) quadratic convergence

(c) Aitken's δ^2-formula

Problem 2: $f(x)=(x-1)/x$ has a zero at $p=1$. Formally show that the iteration $x_{k+1}=g(x_k)$ will converge or diverge for all x_o in some interval about p if

(a) $g(x)=x-\frac{f(x)}{2}$

(b) $g(x)=1+xf(x)$

Problem 3: Use any iterative method with $x_o=2.0$ to find $\sqrt{5}$ correct to three significant figures.

Problem 4: (a) The numerical data given below was produced by a convergent iteration $x_{k+1}=g(x_k)$ where $g(x)$ satisfies the linear iteration theorem. The error at each step is given by $e_k=x_k-p$. On the basis of the numerical results, determine if the convergence is linear, quadratic, or neither. To be correct, you must justify your answer. Estimate $g'(p)$.

k	e_k	e_k/e_{k-1}	e_k/e_{k-1}^2
11	.12150000E+00	.40500000E+00	.13500000E+01
12	.20246568E-01	.17486888E+00	.14392500E+01
13	.67232947E-03	.31644144E-01	.14893767E+01
14	.677888842E-06	.10008268E-02	.14996638E+01

(b) Answer question 4(a) for the following data.

k	e_k	e_k/e_{k-1}	e_k/e_{k-1}^2	$e_k/(e_{k-1}e_{k-2})$
2	-.14695719E-02	.29289691E+00	-.58376306E+02	-.19930716E+03
3	-.43042800E-03	.29289347E+00	-.19930553E+03	-.68047030E+03
4	-.12606945E-03	.29289324E+00	-.68046976E+03	-.23232689E+04
5	-.36924888E-04	.29289322E+00	-.23232688E+04	-.79321356E+04

Problem 5: (a) Write Newton's iteration equations to find a simultaneous solution of $f(x,y)=0, g(x,y)=0$.

(b) Using Newton's method with $(x_0, y_0)=(1.1, .9)$, find an improved estimate (x_1, y_1) to the simultaneous solution of $f(x,y)=x^2-2xy+y^2, g(x,y)=2x+2y-4$.

Examination 2

WORK ALL PROBLEMS. EACH PROBLEM IS WORTH 25 POINTS.

Problem 1: (a) Describe three tests for an ill-conditioned system Ax=b.

(b) Suppose the results of elimination on a system Ax=b yields the reduced augmented matrix

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 4 & 8 \\ 0 & 5 & 3 & 2 & 4 \\ 0 & 0 & 10^{-4} & 0 & 11 \\ 0 & 0 & 0 & 10^{-5} & 1 \end{array}\right]$$

where k interchanges of rows took place during elimination. What is $|A|$? Can we conclude that A is ill-conditioned? (Justify your answer.)

Problem 2: (a) Suppose we wish to solve Ax=b by the method of simultaneous displacements (method of Jacobi) where

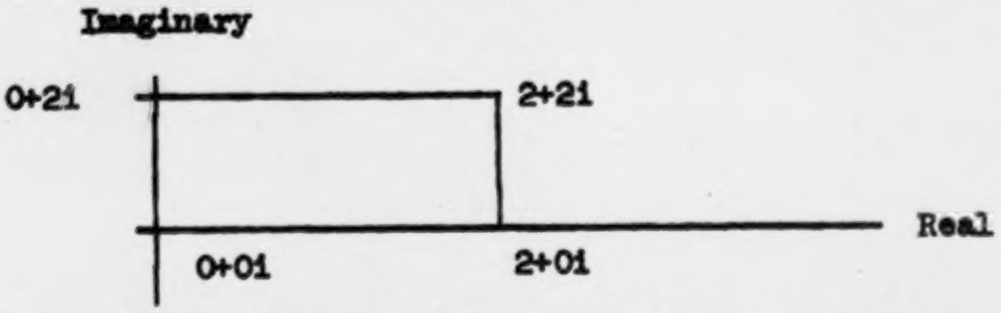
$$A = \begin{bmatrix} a_{11} & 0 & . & . & . & . & 0 & a_{1n} \\ 0 & a_{22} & 0 & . & . & . & 0 & a_{2n} \\ 0 & 0 & a_{33} & 0 & . & . & 0 & a_{3n} \\ . & & & . & & & . & . \\ . & & & & . & & . & . \\ . & & & & & . & . & . \\ 0 & . & . & . & . & . & 0 & a_{nn} \end{bmatrix},$$

that is $a_{ii} \neq 0, a_{in} \neq 0$ for $i=1,\ldots,n$ and all other $a_{ij}=0$. Show that the iteration will converge in a finite number of steps, i.e., $x^{(k)}=x$ for a finite value of k. What is the maximum possible value for k?

(b) For which of the following coefficient matrices A(in Ax=b) can we be guaranteed that the method of simultaneous displacements will converge? (Justify your answer.)

$$A = \begin{bmatrix} 6 & 3 & 0 & 3 \\ 2 & 12 & 4 & 6 \\ 2 & 6 & 18 & 10 \\ 2 & 3 & 14 & 19 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} \\ 1/8 & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix}$$

Problem 3: $p(x)=x^3+x^2+\frac{1}{2}x+5$ has a complex root in the rectangle in the complex plane defined by the vertices 0+0i, 2+0i, 2+2i, 0+2i.



(a) Name another rectangle in the complex plane which contains another complex root of $p(x)$.

(b) Suppose we use an initial approximation to the complex root as $z_0=1+i$. Then the corresponding initial approximate quadratic factor $x^2-\alpha_0 x-\beta_0$ of $p(x)$ is x^2-2x+2. Using this initial approximate quadratic factor $x^2-\alpha_0 x-\beta_0$, we wish to use the Newton-Bairstow method to find an improved quadratic factor $x^2-\alpha_1 x-\beta_1$ where we know α_1 and β_1 are solutions of the system

$$\begin{bmatrix} -b_1(\alpha_0,\beta_0) \\ -b_0(\alpha_0,\beta_0) \end{bmatrix} = \begin{bmatrix} \frac{\partial b_1}{\partial \alpha}(\alpha_0,\beta_0) & \frac{\partial b_1}{\partial \beta}(\alpha_0,\beta_0) \\ \frac{\partial b_0}{\partial \alpha}(\alpha_0,\beta_0) & \frac{\partial b_0}{\partial \beta}(\alpha_0,\beta_0) \end{bmatrix} \begin{bmatrix} \alpha_1 - \alpha_0 \\ \beta_1 - \beta_0 \end{bmatrix}.$$

Compute the values of b_1, b_0, $\frac{\partial b_1}{\partial \alpha}$, $\frac{\partial b_0}{\partial \alpha}$, $\frac{\partial b_1}{\partial \beta}$, $\frac{\partial b_0}{\partial \beta}$, α_1, and β_1. Show all work.

Problem 4: Show all work.

$$\text{Let} \quad A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

(a) Use Gaussian Elimination to find A^{-1}.

(b) Compute the normalized determinant of A.

(c) If $Ax=b$ where $b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, use A^{-1} to solve for x.

Examination 3

WORK ALL PROBLEMS. EACH PROBLEM IS WORTH 20 POINTS.

Problem 1: (a) In terms of limits, define what it means for $D(h)$ to be an $O(h^k)$-approximation to a number A.

(b) Consider the following Taylor expansions. (Assume all derivatives continuous.)

$$f_{-1}=f_0-hf_0'+\frac{h^2}{2}f_0''-\frac{h^3}{6}f_0'''+\dots$$

$$f_0=f_0$$

$$f_1=f_0+hf_0'+\frac{h^2}{2}f_0''+\frac{h^3}{6}f_0'''+\dots$$

What are A and k if

$$D(h)=\frac{f_1-f_0}{h}$$

$$D(h)=\frac{f_1-f_{-1}}{2h}$$

$$D(h)=\frac{f_1-2f_0+f_{-1}}{h^2}$$

Problem 2: Suppose we wish to approximate $\int_1^2 \frac{dx}{x}$ by Simpson's Rule.

(a) Write Simpson's approximation for $2n=4$ subdivisions (i.e., What is the formula using $f(x)=\frac{1}{x}$?). Specify the value of h.

(b) The error in Simpson's rule is given by $E_s(h)=-h^4(b-a)f^{IV}(\eta)/180$ where $a<\eta<b$. Determine bounds on n and h to insure $|E_s(h)|<5\cdot10^{-7}$.

Problem 3: Using the differentiation $D(h)=(f(x_0+h)-f(x_0-h))/2h$ and extrapolation to the limit, approximate $f'(1)$ from the values in the following table.

i	-3	-2	-1	0	1	2	3
x_i	.500	.750	.875	1.000	1.125	1.250	1.500
f_i	2.0000	1.3333	1.1428	1.000	.8888	.8000	.6666

Problem 4: Let $y'=f(x,y)=-2xy$ with initial conditions $y(0)=1$. Use Taylor's algorithm of order 2 and step-size $h=.1$ to approximate $y(.1)$.

Problem 5: (a) Let $p(x)=ax^3+bx^2+cx+d$. If we approximate $\int_a^b p(x)dx$ by Simpson's rule, we get the exact answer. Why? Justify your answer.

(b) The following Romberg integration table is generated for approximations to $\int_1^3 \ln(x)dx$. All calculations are rounded to four significant digits.

h				
2	1.099			
1	1.242	1.290		
.5	1.282	1.295	1.297	
.25	1.292	1.295	1.295	1.294

Using the values from the table, write the value of the trapezoidal approximation to the integral for $N=4$ subdivisions of $[1,3]$. Also, what is the value of Simpson's approximation with $2N=4$ subdivisions of $[1,3]$? The correct answer is $\int_1^3 \ln(x)dx=1.29583\ldots$.

Why is the last diagonal entry in the table worse than the third diagonal entry?

Table 21. Rankings by Exam 1 Scores

Student	1	2	3	4	5	6	7	8	9	10	11	12	13
Score	63	95	43	83	59	54	85	89	85	89	54	63	62
Rank	20	1	35	8	25	31	6	4	7	5	32	21	23

Student	14	15	16	17	18	19	20	21	22	23	24	25	26
Score	77	69	48	70	46	58	60	73	64	56	70	95	63
Rank	10	14	33	12	34	26	24	11	18	27	13	2	22

Student	27	28	29	30	31	32	33	34	35	36	37
Score	55	37	69	68	82	64	55	38	68	56	92
Rank	29	37	15	16	9	19	30	36	17	28	3

Mean=66 Median=64

Table 22. Rankings by Exam 2 Scores

Student	1	2	3	4	5	6	7	8	9	10	11	12	13
Score	88	83	67	80	90	53	80	92	62	80	30	75	42
Rank	4	5	21	6	3	27	7	2	24	8	35	12	31

Student	14	15	16	17	18	19	20	21	22	23	24	25	26
Score	0	36	75	53	51	71	42	73	72	79	74	77	66
Rank	36	34	13	28	30	18	32	15	16	9	14	11	22

Student	27	28	29	30	31	32	33	34	35	36	37
Score	72	41	0	59	70	79	52	64	70	54	93
Rank	17	33	37	25	19	10	29	23	20	26	1

Mean=63 Median=70

Table 23. Rankings by Exam 3 Scores

Student	1	2	3	4	5	6	7	8	9	10	11	12	13
Score	62	86	54	100	95	37	77	86	43	98	22	63	0
Rank	22	9	26	1	3	32	13	10	29	2	35	18	37

Student	14	15	16	17	18	19	20	21	22	23	24	25	26
Score	69	63	49	73	38	63	65	61	62	59	89	93	85
Rank	16	19	27	15	31	20	17	24	23	25	7	4	11

Student	27	28	29	30	31	32	33	34	35	36	37
Score	90	83	74	63	18	93	31	42	34	47	87
Rank	6	13	14	21	36	5	34	30	33	28	8

Mean=64 Median=63

Table 24. Rankings by Average Scores Over Three Exams

Student	1	2	3	4	5	6	7	8	9	10	11	12	13
Score	71	88	55	88	81	48	81	89	63	89	35	67	35
Rank	13	5	27	6	7	31	8	2	20	3	36	15	37

Student	14	15	16	17	18	19	20	21	22	23	24	25	26
Score	49	56	57	65	45	64	56	69	66	65	78	88	71
Rank	30	25	22	17	35	19	26	14	16	18	10	4	12

Student	27	28	29	30	31	32	33	34	35	36	37
Score	72	54	48	63	57	79	46	48	57	52	91
Rank	11	28	33	21	24	9	34	32	23	29	1

Mean=64 Median=64

APPENDIX D

SAMPLE STUDENT OUTPUT

```
LESSON A.  KEYBOARD ORIENTATION

 1.  COMMUNICATION BETWEEN THE STUDENT AND COMPUTER

 THE RIGHT-MOST KEY IN THE SECOND ROW FROM THE TOP IS
 LABELLED (RETURN).  PUSH THE (RETURN) KEY.
#
 BY PUSHING THE (RETURN) KEY, YOU SIGNAL THE COMPUTER THAT YOU HAVE
 COMPLETED YOUR RESPONSE.
 TYPE THE LETTER  A  AND PUSH (RETURN).
#A
 OK.  NOTICE THAT THE COMPUTER REMAINS INACTIVE UNTIL YOU
 PRESS (RETURN).  UPON PRESSING (RETURN), THE COMPUTER CHECKS YOUR
 RESPONSE AND TAKES APPROPRIATE ACTION.
 TYPE ANY STRING OF LETTERS, ONE OF WHICH IS THE LETTER  B
#AOEKBLE
 OK.  THE COMPUTER EXPECTS A RESPONSE FROM THE STUDENT IF IT
 TYPES  #  AND STOPS. NOTICE THE  # ON THE NEXT STOP.
 THEN PUSH (RETURN).
#
 OK.  SUMMARIZING,
      1.  THE COMPUTER EXPECTS A RESPONSE FROM THE STUDENT IF IT
          TYPES  #  AND STOPS.
      2.  THE STUDENT SIGNALS COMPLETION OF A RESPONSE BY
          PUSHING (RETURN).

 PUSH (RETURN) TO CONTINUE.
#
 2.  CORRECTION OF TYPING ERRORS - THE (RUBOUT) KEY.

 THE (RUBOUT) KEY ACTS AS A LOGICAL ERASER IN THE SENSE THAT THE
 STUDENT MAY ERASE A PARTIALLY COMPLETED ANSWER AND START ALL OVER.
 THE (RUBOUT) KEY IS ON THE 3RD ROW FROM THE TOP -- RIGHT-HAND
 SIDE.  PUSH (RUBOUT); THEN PUSH (RETURN).
#
#
 OK.  TO SEE HOW (RUBOUT) WORKS, PERFORM THE FOLLOWING STEPS.
      (1) TYPE THE WORD  PLUZ
      (2) PUSH (RUBOUT)
      (3) TYPE THE WORD  PLUS
      (4) PUSH (RETURN)

#PLUZ
#PLUS
 OK.  THE COMPUTER IGNORES EVERYTHING TYPED PRIOR TO PUSHING
 (RUBOUT).  YOUR RESPONSE IS RECOGNIZED AS THE WORD "PLUS",
 JUST AS IF YOU HAD NOT MADE A TYPING ERROR.
 PRESS (RETURN) TO CONTINUE.
```

```
#
 YOU MAY USE THE (RUBOUT) KEY AS OFTEN AS YOU WISH IN ANY
 RESPONSE. SUPPOSE A STUDENT PERFORMS THE FOLLOWING
      (1) TYPES THE WORD  SQUZ
      (2) PUSHES (RUBOUT)
      (3) TYPES THE WORD  SQUARW
      (4) PUSHES (RUBOUT)
      (5) TYPES THE WORD  SQUARE
      (6) PUSHES (RETURN)
 WHAT WOULD THE COMPUTER RECOGNIZE AS THE RESPONSE?
#SQUARE
 OK

 3. CORRECTION OF TYPING ERRORS - THE  #  KEY

 INSTEAD OF USING THE (RUBOUT) KEY TO ERASE A RESPONSE, THE STUDENT
 MAY ELECT TO USE THE  # KEY.  THE  #  KEY IS ON THE TOP ROW ---
  3RD KEY FROM THE LEFT.  WHEN YOU HAVE FOUND IT, PUSH (RETURN).
#
 SINCE THE  #  IS AN UPPER CASE CHARACTER, YOU MUST DEPRESS THE
 (SHIFT) KEY WHEN TYPING  #.  TYPE  #  SEVERAL TIMES.
 PRESS (RETURN) WHEN SATISFIED.
############
 THE  #  ACTS AS A LOGICAL BACKSPACE IN THE SENSE THAT IT
 DELETES BOTH ITSELF AND THE PREVIOUS CHARACTER FROM THE COMPUTER.
 FOR EXAMPLE, SUPPOSE THE STUDENT PERFORMS THE FOLLOWING STEPS
      (1) TYPES ABD
      (2) TYPES  #
      (3) TYPES C
      (4) PUSHES (RETURN)
 WHAT WORD WOULD THE COMPUTER RECOGNIZE AS THE RESPONSE?
#ABC
 OK
 THE  #  KEY MAY BE USED AS OFTEN AS NEEDED IN ANY RESPONSE.
 WHAT WOULD THE COMPUTER RECOGNIZE IF THE STUDENT TYPED X+Y##-Z#W
 AND PUSHED RETURN?
#X-W
 OK
 SUPPOSE THE STUDENT DOES THE FOLLOWING
      (1) TYPES  YES###NOY##O
      (2) PUSHES (RETURN)
 WHAT  ANSWER  WOULD THE COMPUTER RECOGNIZE?
#YES
      NO.  THE  YES###  DELETES THE  YES.  TRY AGAIN.
#NO
 OK.
 WHAT IS THE FINAL ANSWER IF THE STUDENT
      (1) TYPES THE WORD YES##
      (2) TYPES  NO
      (3) PUSHES (RUBOUT)
      (4) TYPES  NO
      (5) PUSHES (RETURN)
#NO
 OK
 THE (RUBOUT) KEY AND # KEY MAY BE USED AT YOUR
 DISCRETION TO CORRECT TYPING ERRORS.
 PRESS (RETURN) TO CONTINUE.
#
```

```
4. MATHEMATICAL EXPRESSIONS

 WITH MINOR EXCEPTIONS, ALL MATHEMATICAL EXPRESSIONS MUST BE
 CONSTRUCTED, USING 'FORTRAN' RULES.
 FIRST, LOCATE THE ARITHMETIC OPERATORS.
 THE  +  KEY IS ON THE SECOND ROW FROM THE BOTTOM--RIGHT SIDE.
 SINCE IT IS AN UPPER CASE LETTER, YOU MUST DEPRESS THE SHIFT KEY
 WHEN YOU PRESS THE PLUS KEY.
 TYPE THE CHARACTER  +  SEVERAL TIMES. THEN PRESS (RETURN).
#++++++++
 TYPE THE EXPRESSION  X+Y+Z
#X>#+Y+Z
 OK.
 THE  *  (MULTIPLICATION) IS AN UPPER CASE CHARACTER LOCATED ON THE
 TOP ROW---RIGHT SIDE.
 TYPE THE EXPRESSION X*Y+Z
#X:#*Y+Z
 OK
 THE  -  (MINUS SIGN) IS LOWER CASE LOCATED ON THE TOP ROW---
 RIGHT SIDE.
 TYPE THE EXPRESSION X-Y+Z*W
#X-Y+Z*W
 OK. THE / (DIVISION) IS LOCATED ON THE BOTTOM ROW---RIGHT SIDE
 TYPE THE EXPRESSION X/Y-Z+W*K
#X/Y-Z+W=#*I#K
 OK.  EXPONENTIATION IS DENOTED BY ** OR ↑.
 TYPE THE EXPRESSION X*Y/Z-W**K+A
#X*Y/Z-W**K+A
 OK.  THE ↑ (UP-ARROW) MAY ALSO BE USED TO DENOTE
 EXPONENTIATION AND IS LOCATED ON THE BOTTOM ROW---IN THE MIDDLE.
 TYPE THE EXPRESSION X*Y/Z-W↑K+A
#X*Y/Z-W↑K+A
 OK.

 5.  SUBSCRIPTED VARIABLES

 THE SYMBOLS [ AND ] ARE USED TO DENOTE SUBSCRIPTS.
 FOR EXAMPLE, X[3] DENOTES THE THIRD ELEMENT OF THE ARRAY X.
 YOU WILL FIND THE BRACKET SYMBOLS [ AND ] ON THE FIRST AND SECOND
 ROWS FROM THE BOTTOM.  LOCATE THESE SYMBOLS AND
 TYPE THE NAME FOR THE 15TH ELEMENT OF AN ARRAY CALLED W
#W[15]
 OK.
 REMEMBER THAT BRACKETS (NOT PARENTHESIS) ARE USED TO DENOTE
 SUBSCRIPTS. WHAT IS THE I+1,J-1 ELEMENT OF THE ARRAY A?
#A[I+1,J-1]
 OK.

 6. THE DISTINGUISHED NAME "PI"

 THE NAME "PI" DENOTES THE CONSTANT 3.14159...ETC.
 WHENEVER YOU WISH TO USE THIS CONSTANT, YOU MERELY TYPE THE
 WORD PI.  FOR EXAMPLE, COS(PI)=-1.
 PRESS (RETURN) TO CONTINUE.
#
 WHAT ARE 3 DISTINCT VALUES OF X (IN RADIANS) SO THAT SIN(X)=0?
 ONE VALUE IS X=
#PI
```

```
 OK
 ANOTHER VALUE IS X=
#0
 OK
 A THIRD VALUE IS X=
#194PI
 ILLEGAL CHARACTER OR COMBINATION        4P
 TYPE A CORRECT EXPRESSION.
#194*1#PII#
 OK.

 7. AVAILABLE MATHEMATICAL FUNCTIONS - LATITUDE IN USAGE

 THE FOLLOWING SUBSET OF FORTRAN FUNCTIONS MAY BE USED BY THE
 STUDENT AT ANY TIME.

  (A) TRIGONOMETRIC   - SIN,COS,TAN,SEC,CSC,COTAN
                      - ARCSIN (ASIN), ARCCOS (ACOS), ARCTAN (ATAN)
  (B) HYPERBOLIC      - SINH,COSH,TANH
  (C) SQUARE ROOT     - SQRT
  (D) EXPONENTIAL     - EXP
  (E) NATURAL LOG     - LN OR ALOG
  (F) BASE 10 LOG     - LOG OR ALOG10
  (G) ABSOLUTE VALUE  - ABS
  (H) INTEGER VALUE   - INT

 IN ANSWERING QUESTIONS, YOU ARE USUALLY FREE TO LEAVE YOUR ANSWERS
 IN UNREDUCED FORM.  FOR EXAMPLE, 1/100, .01, AND 10**(-3) ARE
 EQUIVALENT ANSWERS AND ARE RECOGNIZED AS SUCH.  LIKEWISE,
 ALGEBRAIC EXPRESSIONS NEED NOT BE REDUCED. FOR EXAMPLE, X**3/X
 AND X**2 ARE THE SAME. COS(X)**2, 1-SIN(X)**2, 1/SEC(X)**2, AND
 2*COS(X)**2-1+SIN(X)**2 ARE RECOGNIZED AS EQUIVALENT.  WHENEVER
 THE ARITHMETIC OR ALGEBRA BECOMES BURDENSOME, LEAVE YOUR ANSWER IN
 AN EQUIVALENT FORM.
 PRESS (RETURN) TO CONTINUE
#

 8.  FIRST ESTIMATES OF ZEROS (ROOTS) OF FUNCTIONS

 THE REMAINDER OF THIS LESSON IS DEVOTED TO LEARNING SEVERAL
 SIMPLE TECHNIQUES TO ESTIMATE THE LOCALITY OF ZEROS (ALSO CALLED
 ROOTS) OF A GIVEN FUNCTION F(X).
 PRESS (RETURN) TO BEGIN.
#
 A.  METHOD 1 - CHANGE OF SIGN OF F(X)

 IF F(A)>0 AND F(B)<0 (OR VISA VERSA) AND F(X) IS CONTINUOUS ON THE
 INTERVAL [A,B], THEN F(X) HAS A ZERO IN THE INTERVAL [A,B],
 THAT IS, THERE IS A NUMBER P SO THAT A<P<B AND F(P)=0.  A SIMPLE
 WAY TO DESCRIBE THE CHANGE OF SIGN IS TO SAY F(A)*F(B)<0.
 PRESS (RETURN) FOR EXAMPLES.
#
 EXAMPLE A1.  LET F(X)=X-EXP(-X).  ON WHICH OF THE FOLLOWING
 INTERVALS DOES F(X) CHANGE SIGN?
   (A)  [A,B]=[-2,-1]
   (B)  [A,B]=[-1,0]
   (C)  [A,B]=[0,1]
   (D)  [A,B]=[1,2]
   (E)  [A,B]=[0,2]
 CORRECT ANSWER IS (A,B,C,D,E)?
#D
```

```
     NO.  F(A)*F(B)>0 WHICH GIVES NO INDICATION OF A ZERO IN THE
     INTERVAL [A,B].  TRY AGAIN OR TYPE HELP.
 CORRECT ANSWER IS (A,B,C,D,E)?
#HELP
 FOR EACH OF THE ABOVE INTERVALS, CHECK F(A)*F(B)<0.
 CORRECT ANSWER IS (A,B,C,D,E)?
#E
 OK.  F(0)=0-EXP(0)=-1<0, F(2)=2-EXP(-2)=2-1/EXP(2)>0.  SO
 F(0)*F(2)<0 AND F(X) CHANGES SIGN ON [0,2].  A BETTER CHOICE,
 HOWEVER, WOULD HAVE BEEN THE INTERVAL (0,1) SINCE IT AFFORDS A
 BETTER ESTIMATE OF THE ROOT OF F(X).
 PRESS (RETURN) TO CONTINUE.
#
 EXAMPLE A2.  LET F(X)=X+2-EXP(-X).  FIND AN INTERVAL [A,B] SO THAT
 F(A)*F(B)<0.  WHEN READY, PUSH (RETURN).
#
 A=
#1
 B=
#-1
     CHOOSE B>A.  START ALL OVER.
 A=
#0
 B=
#10
     NO. F(A)*F(B)>0.  TRY AGAIN.
 A=
#0
 B=
#1/2
     NO. F(A)*F(B)>0.  TRY AGAIN.
 A=
#0
 B=
#1/4
     NO.  [A,B]=[-2,0] IS ACCEPTABLE SINCE F(-2)=-2+2-EXP(2)<0,
     F(0)=2-EXP(0)=2-1=1>0, AND F(-2)*F(0)<0.  PRESS (RETURN).
#
 DO YOU WANT TO TRY A SMALLER INTERVAL FOR THIS EXAMPLE?
#YES
 A=
#-1
 B=
#0
 OK
 DO YOU WANT TO TRY A SMALLER INTERVAL FOR THIS EXAMPLE?
#YES
 A=
#-1
 B=
#-1/2
     NO. F(A)*F(B)>0.  TRY AGAIN.
 A=
#-1/2
 B=
#0-#
 OK
 DO YOU WANT TO TRY A SMALLER INTERVAL FOR THIS EXAMPLE?
#NO
```

```
EXAMPLE A3.  LET F(X)=2*X-TAN(X).  BY GRAPHING F(X), FIND AN
INTERVAL [A,B] WHICH CONTAINS THE "SMALLEST POSITIVE ROOT" OF F(X).
WHEN READY, PRESS (RETURN).
#
 A=
#PI
       YOU HAVE CHOSEN A>PI/2.  NOTE THAT F(X) HAS A ROOT LESS THAN
       PI/2.  TRY AGAIN.
 A=
#0
       SINCE WE SEEK A POSITIVE ROOT, YOU WOULD NATURALLY CHOOSE A>0.
 A=
#PI/4
 B=
#2*PI/3
       F(X) IS DISCONTINUOUS AT PI/2.  FIND AN INTERVAL WHERE F(X)
       IS CONTINUOUS.
 B=
#PI/3
       NO.  F(A)*F(B)>0.  TRY AGAIN.
 A=
#PI/4
 B=
#PI/2-.0000001
 OK
 DO YOU WISH TO FIND A SMALLER INTERVAL FOR THIS EXAMPLE?
#NO
 REMARK---ESTIMATING THE ROOT OF F(X) BY FINDING AN INTERVAL [A,B]
 SO THAT F(A)*F(B)<0 WILL WORK IF F(X) HAS A SIMPLE ROOT.  CONSIDER
 THE EXAMPLE F(X)=X**2.  FOR ALL X DIFFERENT FROM 0, F(X)>0 AND IT
 IS IMPOSSIBLE TO APPLY THE TEST.  THE SAME IS TRUE FOR
 COS(X)**2.   PRESS (RETURN) FOR ANOTHER METHOD.
#

 B.  METHOD 2 - INTERSECTION OF FUNCTIONS

 ANOTHER TECHNIQUE IS TO SET F(X)=0 AND TRANSFORM THE PROBLEM INTO
 G(X)=H(X) WHERE IT IS EASIER TO GRAPH G(X) AND H(X).

 EXAMPLE B1.  LET F(X)=X-EXP(-X).  SETTING F(X)=0, WE HAVE THE
 EQUATION X-EXP(-X)=0.  ADDING EXP(-X) TO BOTH SIDES GIVES
 X=EXP(-X).  SO G(X)=X AND H(X)=EXP(-X).   TO ESTIMATE A ROOT OF
 F(X) IS EQUIVALENT TO ESTIMATING A POINT OF INTERSECTION OF THE
 FUNCTIONS G(X) AND H(X).  PRESS (RETURN) FOR A ROUGH GRAPH.
#
```

```
BY PROPER MANIPULATION,                  *   !         .
WE TRANSFORM THE                H(X)=EXP(-X) !        .
PROBLEM OF GRAPHING                       *  !    G(X)=X
F(X) INTO ONE OF GRAPHING                    !      .
TWO SIMPLE FUNCTIONS, G(X)=X                *!     .
AND H(X)=EXP(-X).  WE ESTIMATE               !    .
THE INTERSECTION P OF G(X) AND               ! * .
AND H(X) AND ESTIMATE (A,B) SO               !  . *
A<P<B.  MORE PRECISELY, WE SEEK              ! .      *
[A,B] SO THAT EITHER                         !.             *
                                        -----0---P-----1---------
  (1) H(A)>G(A) AND H(B)<G(B) OR             !
  (2) H(A)<G(A) AND H(B)>G(B).               !

 IF WE CHOOSE A=0 AND B=1, WE FIND H(0)=1>0=G(0) AND
 H(1)=1/EXP(1)<1=G(1).  SO THE FIRST CASE HOLDS AND WE HAVE [A,B].
 PRESS (RETURN) TO CONTINUE.
#
 EXAMPLE B2.  LET F(X)=X-COS(X).  NAME TWO FUNCTIONS G(X) AND H(X)
 WHICH INTERSECT AT A ROOT OF F(X).
 G(X)=
#X
 H(X)=
#COS(X)
 OK
 DO YOU WANT TO TRY DIFFERENT FUNCTIONS G(X) AND H(X)?
#NO
 GRAPH YOUR LAST G(X) AND H(X) TO ESTIMATE THE INTERSECTION P OF
 G(X) AND H(X).  THEN CHOOSE (A,B) SO A<P<B.
 WHEN READY, PRESS (RETURN).
#
 A=
#0
 B=
#1
 OK
 EXAMPLE B3.  LET F(X)=X**3+X**2-2*X-2.  DEFINE G(X) AND H(X)
 WHICH INTERSECT AT A ROOT OF F(X).  PRESS (RETURN) WHEN READY.
#
 G(X)=
#-X**3
 H(X)=
#X**2-2*X-2
 OK

 C.  METHOD 3 - TECHNIQUES FOR REAL ROOTS OF POLYNOMIALS

 CONSIDER THE GENERAL FORM OF A POLYNOMIAL AS

   P(X)=A[N]*X**(N)+A[N-1]*X**(N-1)+...+A[1]*X+A[0].

 IF SOME ROOT P IS LARGE IN ABSOLUTE VALUE WITH COMPARISON TO THE
 OTHERS, THEN THE FIRST TWO TERMS A[N]*X**(N)+A[N-1]*X**(N-1)
 ARE DOMINANT AND THE SOLUTION TO A[N]*X**(N)+A[N-1]*X**(N-1)
 "MAY" YIELD A "GOOD" ESTIMATE.  EQUIVALENTLY (FACTORING OUT
 X**(N-1)), WE SEEK A ROOT OF Q(X)=A[N]*X+A[N-1] AS THE ESTIMATE
 OF THE ROOT OF P(X).  PRESS (RETURN) FOR EXAMPLES.
#
```

```
 EXAMPLE C1.  LET P(X)=X**3-11.1*X**2+11.1*X-1.  A FIRST ESTIMATE
 OF THE LARGEST ROOT (IN MAGNITUDE) OF P(X) IS GIVEN BY THE ZERO OF
 Q(X)=
#X-11.1
 OK
 THIS GIVES 11.1 AS AN ESTIMATE TO THE LARGEST ROOT IN
 MAGNITUDE OF P(X).  THE LARGEST ROOT IS ACTUALLY 10.
 PRESS (RETURN) TO CONTINUE.
#
 EXAMPLE C2.  LET P(X)=X**4+10*X**3-11.2*X**2-2*X+2.2   USING
 THE TECHNIQUE ABOVE, AN ESTIMATE OF THE LARGEST ROOT IN ABSOLUTE
 VALUE IS WHAT NUMBER?
 ESTIMATE=
#10
      TRY AGAIN OR TYPE HELP.
 ESTIMATE=
#HELP
      TAKE THE FIRST TWO TERMS OF P(X) AND FACTOR OUT THE X**3.
      SET THE RESULT TO 0 AND SOLVE FOR X.
 ESTIMATE=
#-10
 OK
 SO -10 IS AN ESTIMATE OF THE LARGEST ROOT (IN ABSOLUTE VALUE) OF
 P(X).  THE LARGEST ROOT IS ACTUALLY -11.   PRESS (RETURN).
#
 IN ORDER TO ESTIMATE THE SMALLEST ROOT IN ABSOLUTE VALUE, THE
 LAST TWO TERMS, NAMELY A[1]*X+A[0], ARE USED.  SOLVING
 A[1]*X+A[0]=0 "MAY" PROVIDE A "GOOD" ESTIMATE.  PRESS (RETURN).
#
 EXAMPLE C3.  CONSIDER AGAIN P(X)=X**3-11.1*X**2+11.1*X-1.  FIND AN
 ESTIMATE TO THE SMALLEST ROOT (IN ABSOLUTE VALUE) BY THE ABOVE
 TECHNIQUE.
 ESTIMATE=
#1/11.1
 OK
 THE SMALLEST ROOT OF P(X) IS ACTUALLY P=.1.  PRESS (RETURN).
#
 THIS IS THE END OF LESSON A.

 YOU HAVE TWO CHOICES

 (1)  TYPE "$LOGOFF" WHICH WILL SIGN YOU OFF THE COMPUTER.
 (2)  TYPE "$LESSON,-----" WHERE ----- IS THE SECTION NAME
      OF THE MATERIAL YOU WISH TO STUDY.

 CONSULT THE INDEX OF THE STUDENT MANUAL FOR THE LIST OF
 AVAILABLE SECTION NAMES.

#$LOGOFF
```

```
 LESSON 10.   THE NEWTON-BAIRSTOW METHOD FOR FINDING COMPLEX ZEROS
              OF A POLYNOMIAL.    PRESS (RETURN) TO BEGIN.
#
 1.  REVIEW OF THE DIVISION ALGORITHM FOR DIVIDING
 P(X)=A[N]*X**N+A[N-1]*X**(N-1)+...+A[0] BY X**2-S*X-T.

 THE DIVISION ALGORITHM (LESSON 9) IS P(X)=Q(X)*(X**2-S*X-T)+R(X)
 WHERE Q(X)=B[N]*X**(N-2)+...+B[3]*X+B[2] AND R(X)=B[1]*(X-S)+B[0]
 AND THE B[I] ARE COMPUTED BY THE RECURSION FORMULAS B[N]=A[N],
 B[N-1]=A[N-1]+S*B[N], B[I]=A[I]+S*B[I+1]+T*B[I+2], I=N-2,...,1,0.
 PRESS (RETURN) TO CONTINUE.
#
 EXAMPLE 1A. LET P(X)=X**4-2*X**3+X**2+2*X-2.  IT IS KNOWN THAT
 P(X) HAS A COMPLEX ROOT NEAR Z=1+.9*I. SINCE P(X) HAS REAL
 COEFFICIENTS, ANOTHER COMPLEX ROOT IS NEAR ZBAR=1-.9*I, THAT IS,
 AN APPROXIMATE QUADRATIC FACTOR IS (X-Z)*(X-ZBAR).
 WRITE THE APPROXIMATE QUADRATIC FACTOR IN TERMS OF X AND REAL
 NUMBERS.
 (X-Z)*(X-ZBAR)=
#X**2-1.81*X
      TRY AGAIN OR TYPE HELP.
 (X-Z)*(X-ZBAR)=
#HELP
      (X-Z)*(X-ZBAR)=X**2-(Z+ZBAR)*X-Z*ZBAR.  Z+ZBAR AND Z*ZBAR
      ARE REAL.  CARRY OUT THE MULTIPLICATION TO ELIMINATE "I".
 (X-Z)*(X-ZBAR)=
#X↑2-2*X-1.81
      TRY AGAIN OR TYPE HELP.
 (X-Z)*(X-ZBAR)=
#X**2-2*X-1.81
      (X-Z)*(X-ZBAR)=(X-(1+.9*I))*(X-(1-.9*I))=X**2-2*X+1.81
      PRESS (RETURN) TO CONTINUE.
#
 WE WILL CONSIDER THIS APPROXIMATE QUADRATIC FACTOR X**2-2*X+1.81
 AS AN INITIAL ESTIMATE TO THE TRUE FACTOR AND DENOTE IT BY
 X**2-S[0]*X-T[0].  WHAT ARE THE VALUES OF S[0] AND T[0]?
 S[0]=
#2
 OK
 T[0]=
#-1.81
 OK
 WE DIVIDE P(X)=X**4-2*X**3+X**2+2*X-2 BY X**2-S[0]*X-T[0]
 OR X**2-2*X+1.81 TO FORM P(X)=Q(X)*(X**2-S[0]*X-T[0])+R(X).
 WRITE THE RECURSION FORMULAS, NUMERICAL EXPRESSIONS, OR VALUES FOR
 THE B[I] TO DEFINE Q(X)=B[4]*X**2+B[3]*X+B[2] AND
 R(X)=B[0]*(X-S[0])+B[0].
 B[4]=
#1
 OK.  B[4]=A[4]=1.
 B[3]=
#-2+S*B[4]
```

```
      TRY AGAIN OR TYPE HELP.
 B[3]=
#0
 OK, B[3]=A[3]+S[0]*B[4]=-2+2*1=0.
 B[2]=
#1+S[0]*B[3]+T[0]*B[4]
 OK,  B[2]=A[2]+S[0]*B[3]+T[0]*B[4]=1+2*0-1.81*1=-.81.
 B[1]=
#2-.81*S[0]
 OK, B[1]=A[1]+S[0]*B[2]+T[0]*B[3]=2+2*(-.81)-1.81*0=.38
 B[0]=
#-2+.38*S[0]-.81*T[0]
 OK, B[0]=A[0]+S[0]*B[1]+T[0]*B[2]=-2+2*.38-1.81*(-.81)=.2261
 SO P(X)=Q(X)*(X**2-S[0]*X-T[0])+R(X) WHERE
 Q(X)=
#HELP
      Q(X)=B[4]*X**2+B[3]*X+B[2]=X**2-.81.  PRESS (RETURN).
#
 R(X)=
#.38*X+.2261
      R(X)=B[1]*(X-S[0])+B[0]=.38*(X-2)+.2261. PRESS (RETURN).
#
 NOTE THAT R(X) IS NOT ZERO IN THIS EXAMPLE SINCE
 X**2-S[0]*X-T[0] IS NOT AN EXACT DIVISOR OF P(X).  THE NEWTON-
 BAIRSTOW METHOD PROVIDES A WAY TO IMPROVE THE DIVISOR BY FORCING
 R(X) TO ZERO.  PRESS (RETURN) TO CONTINUE.
#
 2.  THE NEWTON-BAIRSTOW METHOD FOR IMPROVING AN APPROXIMATE
     QUADRATIC FACTOR

 THE NEWTON-BAIRSTOW METHOD AMOUNTS TO SETTING
      R(X)=B[1]*(X-S)+B[0] = 0
 AND SOLVING FOR IMPROVED VALUES OF S AND T BY NEWTON'S METHOD.
 SETTING R(X)=0 IS EQUIVALENT TO SETTING
      B[1]=A[1]+S*B[2]+T*B[3]=0
      B[0]=A[0]+S*B[1]+T*B[2]=0
 THIS SYSTEM IS SOLVED BY NEWTON'S METHOD FOR SYSTEMS (LESSON 7).
 PRESS (RETURN) FOR NEWTON'S EQUATIONS.
#
 NEWTON'S EQUATIONS WERE DERIVED IN LESSON 7 AS
      [B1'S]*(S-S[0])+[B1'T]*(T-T[0])=-B[1]
      [B0'S]*(S-S[0])+[B0'T]*(T-T[0])=-B[0]
 IN ORDER TO SOLVE THIS SYSTEM FOR S AND T, WE MUST DETERMINE THE
 VALUES FOR THE PARTIAL DERIVATIVES OF B[1] AND B[0] EVALUATED AT
 S[0] AND T[0].  THESE PARTIALS ARE DENOTED BY [B1'S], [B1'T],
 [B0'S], AND [B0'T].  PRESS (RETURN) FOR EXAMPLE.
#
```

```
 EXAMPLE 2A. LET P(X)=A[4]*X**4+A[3]*X**3+A[2]*X**2+A[1]*X+A[0].
 TO DETERMINE Q(X) AND R(X) FOR AN APPROXIMATE QUADRATIC FACTOR
 X**2-S*X-T, WE COMPUTE B[4]=A[4], B[3]=A[3]+S*B[4],
 B[2]=A[2]+S*B[3]+T*B[4], B[1]=A[1]+S*B[2]+T*B[3], AND
 B[0]=A[0]+S*B[1]+T*B[2].  THIS GIVES THE VALUES OF B[1] AND B[0]
 NEEDED FOR THE RIGHT SIDE OF NEWTON'S EQUATIONS.  WE NEXT
 CALCULATE [B1'S] AND [B0'S] BY RECURSION.
 TAKE THE PARTIAL DERIVATIVE OF B[4] WITH RESPECT TO S.
 C[5]=[B4'S]=
#HELP
      B[4]=A[4] WHICH IS CONSTANT.  TAKE THE DERIVATIVE WITH
      RESPECT TO S.  WE CALL THE RESULT C[5].
 C[5]=[B4'S]=
#0
 OK
 NEXT, TAKE THE DERIVATIVE OF B[3]=A[3]+S*B[4] WITH RESPECT TO S.
 C[4]=[B3'S]=
#B[4]
 OK
 USING THE SYMBOL C[4] FOR [B3'S] AND RECALLING [B4'S]=0, COMPUTE
 C[3]=[B2'S]=
#C[3]+B[3]
      TRY AGAIN OR TYPE HELP.
 C[3]=[B2'S]=
#2*C[3]+B[4]
      B[2]=A[2]+S*B[3]+T*B[4].  DENOTE [B3'S] BY C[4], NOTE T'S=0,
      [B4'S]=0, AND A[2] IS CONSTANT.  SO
      [B2'S]=0+S*[B3'S]+1*B[3]+T*[B4'S]+[T'S]*B[4]=B[3]+S*B[4].
      PRESS (RETURN) TO CONTINUE.
#
 USING  THE SYMBOLS C[3] FOR [B2'S] AND C[4] FOR [B3'S], COMPUTE
 C[2]=[B1'S]=
#HELP
      B[1]=A[1]+S*B[2]+T*B[3].  DENOTE [B2'S] BY C[3] AND [B3'S] BY
      [C4]. NOTE THAT T'S=0 AND A[1] IS CONSTANT.
 C[2]=[B1'S]=
#2*S*B[3]+S*B[4]+S*B[3]
      TRY AGAIN OR TYPE HELP.
 C[2]=[B1'S]=
#0+C[3]+B[4]
      B[1]=A[1]+S*B[2]+T*B[3].  DENOTE [B2'S] BY C[3] AND [B3'S] BY
      C[4].  NOTE T'S=0 AND A[1] IS CONSTANT.  SO
      [B1'S]=0+S*[B2'S]+1*B[2]+T*[B3'S]+[T'S]*B[3]  OR
        C[2]=B[2]+S*C[3]+T*C[4].    PRESS (RETURN) TO CONTINUE.
#
 USING THE SYMBOLS C[2] FOR [B1'S] AND C[3] FOR [B2'S], COMPUTE
 C[1]=[B0'S]=
#S*C[2]+B[1]+T*C[3]
```

```
 OK
 SO, TO FIND THE VALUES OF [B1'S] AND [B0'S] FOR NEWTON'S
 EQUATIONS, WE COMPUTE C[4]=B[4], C[3]=B[3]+S*C[4],
 C[2]=[B1'S]=B[2]+S*C[3]+T*C[4], C[1]=[B0'S]=B[1]+S*C[2]+T*C[3].
 TO FIND THE VALUES [B1'T] AND [B0'T], WE PROCEED IN A SIMILAR
 MANNER.  PRESS (RETURN) TO CONTINUE.
#
 TO COMPUTE [B1'T] AND [B0'T] FOR NEWTON'S METHOD, WE START BY
 COMPUTING [B4'T] AND CALLING THE RESULT D[6].  COMPUTE
 D[6]=[B4'T]=
#0
 OK
 TAKE THE DERIVATIVE WITH RESPECT TO T OF B[3]=A[3]+S*B[4].
 D[5]=[B3'T]=
#0
 OK
 TAKE THE DERIVATIVE WITH RESPECT TO T OF B[2]=A[2]+S*B[3]+T*B[4]
 D[4]=[B2'T]=
#B[4]
 OK
 USING THE SYMBOL D[4] FOR [B2'T], COMPUTE THE PARTIAL WITH
 RESPECT TO T OF B[1]=A[1]+S*B[2]+T*B[3]
 D[3]=[B1'T]=
#T*D[4]+B[3]+S*B[4]
      TRY AGAIN OR TYPE HELP.
 D[3]=[B1'T]=
#HELP
      A[1] IS CONSTANT.  [S'T]=[B3'T]=0. DENOTE [B2'T] BY D[4].
 D[3]=[B1'T]=
#B[3]+S*D[4]
 OK
 USING THE SYMBOLS D[3] FOR [B1'T] AND D[4] FOR [B2'T]  COMPUTE
 THE DERIVATIVE WITH RESPECT TO T OF B[0]=A[0]+S*B[1]+T*B[2].
 D[2]=[B0'T]=
#
#S*D[3]+B[2]+T*D[4]
 OK
 SO, TO FIND THE VALUES OF [B1'T] AND [B0'T] FOR NEWTON'S
 EQUATIONS, WE COMPUTE D[4]=B[4], D[3]=B[3]+S*D[4]=[B1'T], AND
 D[2]=B[2]+S*D[3]+T*D[4]=[B0'T].  PRESS (RETURN) TO CONTINUE.
#
 NOTE THAT D[4]=C[4], D[3]=C[3], AND D[2]=C[2] SO THAT COMPUTATION
 OF THE D[I] ARE NOT NECESSARY.  USING THE VALUES [B1'T]=C[3]
 [B0'T]=[B1'S]=C[2], AND [B0'S]=C[1], NEWTON'S SYSTEM BECOMES
      C[2]*(S-S[0])+C[3]*(T-T[0])=-B[1]
      C[1]*(S-S[0])+C[2]*(T-T[0])=-B[0]
 PRESS (RETURN) TO CONTINUE.
#
```

```
BY MULTIPLYING THE FIRST OF THESE EQUATIONS BY C[2] AND THE SECOND
BY C[1] AND ELIMINATING T, THE IMPROVED VALUE S IS
     S=S[0]+(C[3]*B[0]-C[2]*B[1])/(C[2]**2-C[1]*C[3])
SIMILARLY, THE IMPROVED VALUE OF T IS
     T=T[0]+(C[1]*B[1]-C[2]*B[0])/(C[2]**2-C[1]*C[3])
AND THE IMPROVED ESTIMATE OF THE QUADRATIC FACTOR IS X**2-S*X-T.
PRESS (RETURN) TO CONTINUE.
#
EXERCISE 3B.  P(X)=X**4-20*X**3+199*X**2+20*X-200 HAS A COMPEX
ROOT NEAR Z=(9+10*I).  COMPUTE THE APPROXIMATE QUADRATIC FACTOR
(X**2-S[0]*X-T[0])=(X-Z)*(X-ZBAR).  WRITE THE NUMERIC VALUES FOR
S[0] AND T[0]
S[0]=
#-18
     TRY AGAIN OR TYPE HELP.
S[0]=
#18
OK
T[0]=
#181
     TRY AGAIN OR TYPE HELP.
T[0]=
#-181
OK
RECALL NEWTON'S EQUATIONS
     C[2]*(S-S[0])+C[3]*(T-T[0])=-B[1]
     C[1]*(S-S[0])+C[2]*(T-T[0])=-B[0]
WHERE WE COMPUTE B[0],B[1] BY THE RECURSION B[N]=A[N],
B[N-1]=A[N-1]+S[0]*B[N],B[I]=A[I]+S[0]*B[I+1]+T[0]*B[I+2] FOR
I=N-2,...,0.  C[1] AND C[2] ARE COMPUTED BY THE RECURSION
C[N]=B[N],C[N-1]=B[N-1]+S[0]*C[N],C[I]=B[I]+S[0]*C[I+1]+T[0]*C[I+2]
FOR I=N-2,...,1.   PRESS (RETURN) WHEN READY TO COMPUTE B[0],B[1],
C[1], AND C[2].
#
B[4]=
#1
OK
B[3]=
#-20+18*1
OK, B[3]=-2.
B[2]=
#199-36
      TRY AGAIN OR TYPE HELP.
B[2]=
#199-36-181
OK, B[2]=-18.
B[1]=
#20-18*18-2*8#(-181
UNBALANCED PARENTHESIS OR BRACKETS
TYPE A CORRECT EXPRESSION.
#20-18*18-2*(-181)
OK, B[1]=58.
B[0]=
#-200+18*58+(-181)*(-18)
```

```
 OK, B[0]=4102.
 WE NOW HAVE THE VALUES FOR THE RIGHT HAND SIDE OF NEWTON'S
 EQUATIONS, NAMELY, -B[0]=-4102 AND -B[1]=-58.  WE MUST YET
 COMPUTE THE VALUES OF THE PARTIALS ,NAMELY, C[3], C[2], AND C[1].
 WRITE THE APPROPRIATE EXPRESSIONS OR VALUES FOR THE C[I].
 C[4]=
#0
      TRY AGAIN OR TYPE HELP.
 C[4]=
#1
 OK, C[4]=1
 C[3]=
#-2+18
 OK, C[3]=16.
 C[2]=
#-18+16*18-181
 OK, C[2]=89
 C[1]=
#58+18*89-181*16
 OK, C[1]=-1236.
 WE NOW HAVE THE VALUES OF THE PARTIALS AND NEWTON'S EQUATIONS ARE
 C[2]*(S-S[0])+C[3]*(T-T[0])=-B[1] OR    89*(S-18)+16*(T+181)=-58
 C[1]*(S-S[0])+C[2]*(T-T[0])=-B[0] OR -1236*(S-18)+89*(T+181)=-4102
 WRITE THE EQUATIONS OR VALUES FOR S AND T WHICH SATISFY THIS
 SYSTEM.
 S=
#
#
#(-58-16*(T+181))/89+18
 THE SOLUTION OF NEWTON'S EQUATIONS GIVES
 S=S[0]+(C[3]*B[0]-C[2]*B[1])/(C[2]**2-C[1]*C[3])
  =18+(16*4102-89*58)/(89**2-(-1236)*16))=18+60470/27697 OR
 S=   .20183269E+02
 PRESS (RETURN) TO CONTINUE.
#
 T=
#T[0]+(C[2]*B[0]-C[1]*B[1])/(C[1]**2-C[2]##3]*C[2])
 THE SOLUTION OF NEWTON'S EQUATIONS GIVES
 T=T[0]+(C[1]*B[1]-C[2]*B[0])/(C[2]**2-C[1]*C[3])
  =-181+(-1236*58-89*4102)/27697=-181-436766/27697 OR
 T=  -.19676943E+03
 PRESS (RETURN) TO CONTINUE.
#
```

```
 THE IMPROVED VALUES S AND T ARE DENOTED BY S[1] AND T[1] AND THE
 IMPROVED QUADRATIC FACTOR IS
      X**2-(  .20183269E+02)*X+  .19676943E+03
 WE NOW REPEAT THE ITERATION WITH THE NEW QUADRATIC FACTOR.
 STEP 1.  COMPUTE B[4]=A[4], B[3]=A[3]+S[K]*B[4],
          B[I]=A[I]+S[K]*B[I+1]+T[K]*B[I+2] FOR I=2,1,0.
 STEP 2.  COMPUTE C[4]=B[4], C[3]=B[3]+S[K]*C[4],
          C[I]=B[I]+S[K]*C[I+1]+T[K]*B[I+2] FOR I=2,1.
 STEP 3.  SOLVE FOR S AND T IN NEWTON'S EQUATIONS
          C[2]*(S-S[K])+C[3]*(T-T[K])=-B[1]
          C[1]*(S-S[K])+C[2]*(T-T[K])=-B[0]
 STEP 4.  SET S[K+1]=S, T[K+1]=T.
 EACH TIME (RETURN) IS PRESSED, STEPS 1-4 WILL BE PERFORMED.
 TYPE 'STOP' WHEN YOU ARE SATISFIED WITH THE CONVERGENCE OF THE
 QUADRATIC FACTOR X**2-S[K]*X-T[K].
#
   I      A[I]              B[I]              C[I]

   4   .10000000E+01    .10000000E+01    .10000000E+01
   3  -.20000000E+02    .18326895E+00    .20366538E+02
   2   .19900000E+03    .59295329E+01    .22022341E+03
   1   .20000000E+02    .10361563E+03    .54093184E+03
   0  -.20000000E+03    .72455131E+03
 K= 2 S[K]=   .19968176E+02 T[K]=  -.19953118E+03
 QUADRATIC FACTOR=X**2- .19968176E+02*X+ .19953118E+03

#
   I      A[I]              B[I]              C[I]

   4   .10000000E+01    .10000000E+01    .10000000E+01
   3  -.20000000E+02   -.31823791E-01    .19936352E+02
   2   .19900000E+03   -.11666411E+01    .19739478E+03
   1   .20000000E+02    .30541433E+01   -.33256013E+02
   0  -.20000000E+03    .93766946E+02
 K= 3 S[K]=   .20000136E+02 T[K]=  -.20000082E+03
 QUADRATIC FACTOR=X**2- .20000136E+02*X+ .20000082E+03

#
   I      A[I]              B[I]              C[I]

   4   .10000000E+01    .10000000E+01    .10000000E+01
   3  -.20000000E+02    .13614781E-03    .20000272E+02
   2   .19900000E+03   -.99809305E+00    .19900926E+03
   1   .20000000E+02    .10773432E-01   -.19847715E+02
   0  -.20000000E+03   -.16510536E+00
 K= 4 S[K]=   .20000000E+02 T[K]=  -.20000000E+03
 QUADRATIC FACTOR=X**2- .20000000E+02*X+ .20000000E+03

#STOP
```

```
 ACCEPTING THE LAST QUADRATIC FACTOR AS AN EXACT DIVISOR OF P(X),
 THE ROOTS ARE
 R1=(    .20000000E+02-SQRT( -.40000000E+03))/2  AND
 R2=(    .20000000E+02+SQRT( -.40000000E+03))/2.
 PRESS (RETURN) TO CONTINUE.
#
 THIS IS THE END OF LESSON 10.
 DO YOU WISH TO BEGIN THE PROBLEM MODE?
#YES

 PROBLEM MODE FOR LESSON 10

 YOU MAY WORK ANY PROBLEMS SPECIFIED IN THE STUDENT MANUAL FOR
 LESSON 10 OR YOU MAY REQUEST THE COMPUTER TO GENERATE A PROBLEM
 FOR YOU.  IF YOU HAVE DIFFICULTY, SOME HELP WILL BE PROVIDED.
 FOR EACH PROBLEM, YOU MUST SPECIFY
      (1) THE RECURSION FORMULAS TO COMPUTE EACH B[I] TO FINALLY
          OBTAIN B[1] AND B[0]
      (2) THE RECURSION FORMULAS TO COMPUTE EACH C[I] TO FINALLY
          OBTAIN C[3], C[2], AND C[1]
      (3) STARTING VALUES S[0] AND T[0] FOR THE INITIAL APPROXIMATE
          QUADRATIC FACTOR BASED ON A GIVEN APPROXIMATE COMPLEX
          ROOT Z.
 THEREAFTER, COMPUTATION IS AUTOMATIC TO SOLVE NEWTON'S EQUATIONS
      C[2]*(S-S[K])+C[3]*(T-T[K])=-B[1]
      C[1]*(S-S[K])+C[2]*(T-T[K])=-B[0]
 WITH S[K+1]=S, T[K+1]=T.
 YOUR PROBLEM SELECTION IS (1,2,3,EXTRA,NONE)?
#1
 PROBLEM 1
 P(X)=X**4+3*X**2+1 HAS A COMPLEX ZERO NEAR Z=0+1.6*I.  USE THE
 NEWTON-BAIRSTOW METHOD TO FIND A QUADRATIC FACTOR OF P(X). THE
 COEFFICIENTS OF P(X) ARE A[4]=1, A[3]=0, A[2]=3, A[1]=0, A[0]=1.
 PRESS (RETURN) WHEN READY TO SPECIFY S[0] AND T[0] FOR THE
 INITIAL APPROXIMATE QUADRATIC FACTOR X**2-S[0]*X-T[0].
#
 S[0]=
#0
 T[0]=
#2.56
      T[0]=-Z*ZBAR EXPRESSED AS A REAL NUMBER.
 T[0]=
#-2.56
 DENOTE THE KTH APPROXIMATION TO THE QUADRATIC FACTOR BY
 X**2-S[K]*X-T[K]. DEFINE THE RECURSION FORMULAS FOR COMPUTING THE
 B[I] USING S[K] AND T[K].  PUSH (RETURN) WHEN READY.
#
 B[4]=
#1
 B[3]=
#0
      B[N-1]=A[N-1]+S[K]*B[N] IS THE SECOND RECURSION FORMULA.
 B[3]=
#S[K]
```

```
 B[2]=
#3+S[K]*B[3]+T[K]*B[4]
 B[1]=
#S[K]*B[2]+T[K]*B[3]
 B[0]=
#1+S[K]*B[1]+T[K]*B[2[#]
 DEFINE THE RECURSION FORMULAS FOR COMPUTING THE C[I] USING S[K]
 AND T[K].  WHEN READY, PUSH (RETURN).
#
 C[4]=
#1
 C[3]=
#
#B[3]+S[K]*C[4]
 C[2]=
#B[2]+S[K]*C[3]+T[K]*C[4]
 C[1]=
#B[1]+S[K]*C[2]+T[K]*C[3]
 NEWTON'S EQUATIONS FOR IMPROVED VALUES S[K+1]=S, T[K+1]=T ARE
      C[2]*(S-S[K])+C[3]*(T-T[K])=-B[1]
      C[1]*(S-S[K])+C[2]*(T-T[K])=-B[0]
 EACH TIME (RETURN) IS PRESSED, THE VALUES OF B[I] AND C[I] AS
 FUNCTIONS OF S[K] AND T[K] (SPECIFIED ABOVE) WILL BE COMPUTED AND
 IMPROVED VALUES S[K+1] AND T[K+1] WILL BE COMPUTED AS THE
 SOLUTION TO NEWTON'S EQUATIONS.  WHEN YOU WISH TO TERMINATE THE
 ITERATION, TYPE 'STOP'.
#
   I      A[I]             B[I]             C[I]

   4   .10000000E+01   .10000000E+01   .10000000E+01
   3  0.              0.              0.
   2   .30000000E+01   .44000000E+00  -.21200000E+01
   1  0.              0.              0.
   0   .10000000E+01  -.12640000E+00
 K= 1 S[K]=  0.              T[K]=  -.26196286E+01
 IMPROVED FACTOR=X**2-( 0.              )*X+(  .26196286E+01)

#
   I      A[I]             B[I]             C[I]

   4   .10000000E+01   .10000000E+01   .10000000E+01
   3  0.              0.              0.
   2   .30000000E+01   .38037736E+00  -.22392453E+01
   1  0.              0.              0.
   0   .10000000E+01   .35548594E-02
 K= 2 S[K]=  0.              T[K]=  -.26180351E+01
 IMPROVED FACTOR=X**2-( 0.              )*X+(  .26180351E+01)

#
```

```
  I       A[I]              B[I]              C[I]

  4   .10000000E+01   .10000000E+01   .10000000E+01
  3  0.              0.              0.
  2   .30000000E+01   .38196488E+00  -.22360702E+01
  1  0.              0.              0.
  0   .10000000E+01   .25202378E-05
 K= 3 S[K]=  0.              T[K]=  -.26180340E+01
 IMPROVED FACTOR=X**2-( 0.                  )*X+(  .26180340E+01)

#
  I       A[I]              B[I]              C[I]

  4   .10000000E+01   .10000000E+01   .10000000E+01
  3  0.              0.              0.
  2   .30000000E+01   .38196601E+00  -.22360680E+01
  1  0.              0.              0.
  0   .10000000E+01   .12505552E-11
 K= 4 S[K]=  0.              T[K]=  -.26180340E+01
 IMPROVED FACTOR=X**2-( 0.                  )*X+(  .26180340E+01)

#STOP
 THIS IS THE END OF PROBLEM 1.  SELECT A NEW PROBLEM.
 YOUR PROBLEM SELECTION IS (1,2,3,EXTRA,NONE)?
#NOM#NE

 YOU HAVE TWO CHOICES

 (1)  TYPE "$LOGOFF" WHICH WILL SIGN YOU OFF THE COMPUTER.
 (2)  TYPE "$LESSON,-----" WHERE ----- IS THE SECTION NAME
      OF THE MATERIAL YOU WISH TO STUDY.

 CONSULT THE INDEX OF THE STUDENT MANUAL FOR THE LIST OF
 AVAILABLE SECTION NAMES.

#$LOGOFF
```

```
 PROBLEM MODE FOR LESSON 3

 YOU MAY WORK ANY PROBLEM SPECIFIED IN THE STUDENT MANUAL FOR
 LESSON 3.  IF YOU HAVE DIFFICULTY, SOME HELP WILL BE PROVIDED.
 THE PROBLEMS WILL BE RESTATED HERE ACCORDING TO THE NUMBER YOU
 SELECT.
 YOUR SELECTION IS (1,2,NONE)
#1
 PROBLEM 1. (CONTE, EX. 2.2-1)
 FIND THE SMALLEST POSITIVE ZERO OF F(X)=2*X-TAN(X) USING LINEAR
 ITERATION  AND AITKEN'S DELTA-SQUARED  FORMULA.  YOU WILL HAVE TO
 SPECIFY
      (A) AITKEN'S DELTA-SQUARED FORMULA
      (B) A CONVERGENT ITERATION FUNCTION G(X)
      (C) AN INTERVAL (A,B) ON WHICH ABS(G'(X))<1
 WHEN READY, PUSH (RETURN).
#
 STATE AITKEN'S ACCELERATION FORMULA
 X'[K]=
#X[K-2]-((X[K-1]-X[K-2])↑2)/X[K]-2*X[K-1]+X[K-2])
 UNBALANCED PARENTHESIS OR BRACKETS
 TYPE A CORRECT EXPRESSION.
#X[K-2]-((X[K-1]-X[K-2])↑2)/X[K]-2*X[K-1]+X[K-2]
      NO.  TRY AGAIN.
 X'[K]=
#X[K-2]-((X[K-1]-X[K-2])↑2)/(X[K]-2*X[K-1]+X[K-2])
 OK
 DEFINE THE ITERATION FUNCTION IN TERMS OF X
 G(X)=
#X+(2*X-TAN(X))/5
 DEFINE THE INTERVAL (A,B)
 A=
#ASIN(.89125)
 B=
#ASIN(.9632)
 DEFINE THE DERIVATIVE
 G'(X)=
#1.4-(SE###((SEC(X)↑2)/5
 UNBALANCED PARENTHESIS OR BRACKETS
 TYPE A CORRECT EXPRESSION.
#1.4-(SEC(X)↑2)/5
      ABS(G'(B))>1
 YOU HAVE 4 CHOICES
      (1) CONTINUE THE PROBLEM ANYWAY
      (2) TERMINATE THE PROBLEM
      (3) REDEFINE G(X)
      (4) REDEFINE (A,B)
 YOUR CHOICE (1,2,3,4)?
#4
 DEFINE THE INTERVAL (A,B)
 A=
#ASIN(.89125)
 B=
#ASIN(.9492)
```

```
 DEFINE THE DERIVATIVE
 G'(X)=
#1.4-(SEC(X)↑2)/5
 CHOOSE A STARTING VALUE
 X[0]=
#ASIN(.9128)
 EACH TIME YOU PRESS (RETURN), THE NEXT ITERATION WILL BE PRINTED.
 WHEN YOU WISH TO BEGIN THE ACCELERATION, TYPE 'AITKEN', AFTER
 WHICH AITKEN'S FORMULA WILL BE AUTOMATICALLY APPLIED AFTER EVERY
 3RD ITERATION.  TO TERMINATE THE ITERATION, TYPE 'STOP'.

   K       X[K]            X'[K]

#
   0   .11500883E+01

#
   1   .11631184E+01

#
   2   .11652669E+01

#
   3   .11655276E+01

#
   4   .11655574E+01

#AITKEN
   5   .11655611E+01   .11655612E+01

   6   .11655612E+01

#
   7   .11655612E+01

#
   8   .11655612E+01   .11655612E+01

   9   .11655612E+01

#STOP
 DO YOU WISH TO TRY A DIFFERENT X[0]?
#YED#S
 X[0]=
#ASIN(.8913)
```

```
EACH TIME YOU PRESS (RETURN), THE NEXT ITERATION WILL BE PRINTED.
WHEN YOU WISH TO BEGIN THE ACCELERATION, TYPE 'AITKEN', AFTER
WHICH AITKEN'S FORMULA WILL BE AUTOMATICALLY APPLIED AFTER EVERY
3RD ITERATION.  TO TERMINATE THE ITERATION, TYPE 'STOP'.

  K       X[K]            X'[K]

#
  0   .11002043E+01

#
  1   .11471354E+01

#
  2   .11625018E+01

#AITKEN
  3   .11655184E+01   .11655651E+01

  4   .11655651E+01

#
  5   .11655616E+01

#
  6   .11655612E+01   .11655612E+01

  7   .11655612E+01

#STOP
 DO YOU WISH TO TRY A DIFFERENT X[0]?
#NO
 DO YOU WISH TO DEFINE A DIFFERENT G(X)?
#NO
 CHOOSE ANOTHER PROBLEM.
 YOUR SELECTION IS (1,2,NONE)
#2
 PROBLEM 2.  (CF. PROBLEM 2, LESSON 2)
 FIND THE POSITIVE ZERO P OF F(X)=.7-X+.3*SIN(X) USING LINEAR
 ITERATION AND AITKEN'S DELTA-SQUARED PROCESS.  YOU WILL HAVE TO
 SPECIFY
      (A) AITKEN'S DELTA-SQUARED FORMULA
      (B) A CONVERGENT ITERATION FUNCTION G(X)
      (C) AN INTERVAL (A,B) ON WHICH ABS(G'(X))<1
 WHEN READY, PUSH (RETURN).
#
 STATE AITKEN'S ACCELERATION FORMULA
 X'[K]=
#X[K-2]-((XX#[K-1]-X[K-2])↑2)/(X[K]-2*X[K-1]+X[K-2])
 OK
 DEFINE THE ITERATION FUNCTION IN TERMS OF X
 G(X)=
#X+(.7-X+.3*SIN(X))/7
```

```
 DEFINE THE INTERVAL (A,B)
 A=
#.5
 B=
#1.5
 DEFINE THE DERIVATIVE
 G'(X)=
#1+(-1+.3*COS(X))/7
 CHOOSE A STARTING VALUE
 X[0]=
#
#1
 EACH TIME YOU PRESS (RETURN), THE NEXT ITERATION WILL BE PRINTED.
 WHEN YOU WISH TO BEGIN THE ACCELERATION, TYPE 'AITKEN', AFTER
 WHICH AITKEN'S FORMULA WILL BE AUTOMATICALLY APPLIED AFTER EVERY
 3RD ITERATION.  TO TERMINATE THE ITERATION, TYPE 'STOP'.

   K        X[K]             X'[K]

#
   0    .10000000E+01

#
   1    .99320590E+00

#
   2    .98722423E+00

#
   3    .98195647E+00

#AITKEN
   4    .97322829E+00    .94295468E+00

   5    .94295468E+00

#
   6    .94293104E+00

#
   7    .94291018E+00    .94275375E+00

   8    .94275375E+00

#
   9    .94275375E+00

#
  10    .94275375E+00    .94275375E+00

  11    .94275375E+00

#STOP
```

```
 DO YOU WISH TO TRY A DIFFERENT X[0]?
#NO
 DO YOU WISH TO DEFINE A DIFFERENT G(X)?
#YES
 DEFINE THE ITERATION FUNCTION IN TERMS OF X
 G(X)=
#X+(.7-X+.3*SIN(X))/20
 DEFINE THE INTERVAL (A,B)
 A=
#.5
 B=
#1.5
 DEFINE THE DERIVATIVE
 G'(X)=
#1+(-1+.3*COS(X))/20
 CHOOSE A STARTING VALUE
 X[0]=
#1
 EACH TIME YOU PRESS (RETURN), THE NEXT ITERATION WILL BE PRINTED.
 WHEN YOU WISH TO BEGIN THE ACCELERATION, TYPE 'AITKEN', AFTER
 WHICH AITKEN'S FORMULA WILL BE AUTOMATICALLY APPLIED AFTER EVERY
 3RD ITERATION.  TO TERMINATE THE ITERATION, TYPE 'STOP'.

   K       X[K]             X'[K]

#
   0   .10000000E+01

#
   1   .99762206E+00

#
   2   .99534372E+00

#AITKEN
   3   .99106903E+00   .94314775E+00

   4   .94314775E+00

#STOp
 DO YOU WISH TO TRY A DIFFERENT X[0]?
#NO
 DO YOU WISH TO DEFINE A DIFFERENT G(X)?
#NO
 CHOOSE ANOTHER PROBLEM.
 YOUR SELECTION IS (1,2,NONE)
#NONE

 YOU HAVE TWO CHOICES

 (1)  TYPE "$LOGOFF" WHICH WILL SIGN YOU OFF THE COMPUTER.
 (2)  TYPE "$LESSON,-----" WHERE ----- IS THE SECTION NAME
      OF THE MATERIAL YOU WISH TO STUDY.

 CONSULT THE INDEX OF THE STUDENT MANUAL FOR THE LIST OF
 AVAILABLE SECTION NAMES.
```

```
PROBLEM MODE AND INVESTIGATION MODE FOR LESSON 12
YOU MAY DEFINE YOUR OWN MATRIX OR REQUEST ONE FROM THE COMPUTER.
FOR EACH PROBLEM, YOU MUST
     (1) SPECIFY THE DIMENSION OF THE MATRIX (2,3,OR 4).
     (2) USE GAUSSIAN ELIMINATION WITH PIVOTING TO REDUCE THE
         AUGMENTED MATRIX TO TRIANGULAR FORM.
     (3) COMPUTE DET(A), AND
     (4) USE BACK-SUBSTITUTION TO COMPUTE B=INVERSE OF A.
YOU MAY TYPE "STOP" AT ANY TIME TO TERMINATE A PROBLEM.
PRESS (RETURN) WHEN READY.
#
 PROBLEM 1.
 SELECT THE SIZE OF MATRIX FOR THIS PROBLEM (2,3,4,NONE).
 N=
#4
 DO YOU WISH TO DEFINE YOUR OWN MATRIX?
#YES
 DEFINE THE ELEMENTS OF YOUR MATRIX ROW-WISE.
 A[1,1]=

#2
 A[1,2]=

#-1
 A[1,3]=

#0
 A[1,4]=

#0
 A[2,1]=

#-1
 A[2,2]=

#2
 A[2,3]=

#-1
 A[2,4]=

#0
 A[3,1]=

#0
 A[3,2]=

#-1
 A[3,3]=

#2
```

```
 A[3,4]=

#-1
 A[4,1]=

#0
 A[4,2]=

#0
 A[4,3]=

#-1
 A[4,4]=

#2
 THE C-MATRIX IS
   .2000000000E+01 -.1000000000E+01 0.                0.
  -.1000000000E+01  .2000000000E+01 -.1000000000E+01 0.
  0.                -.1000000000E+01  .2000000000E+01 -.1000000000E+01
  0.                0.                -.1000000000E+01  .2000000000E+01
 AUGMENTED BY THE IDENTITY MATRIX.
 PERFORM GAUSSIAN ELIMINATION WITH INTERCHANGE BY SPECIFYING THE
 OPERATIONS (1) INTERCHANGE (ROW I) WITH (ROW J)
            (2) REPLACE (ROW J) BY M*(ROW I)+(ROW J)
 AND SUPPLYING THE APPROPRIATE VALUES OF I,J, AND M.
 SPECIFY OPERATION (1,2).
#2
 I=
#1
 J=
#2
 M=
#
#-C[2,1]/C[1,1]
 ROW OPERATION COMPLETE
 THE CURRENT C-MATRIX IS
 ROW 1
  .2000000000E+01 -.1000000000E+01 0.                0.
  .1000000000E+01 0.               0.                0.
 ROW 2
 0.                .1500000000E+01 -.1000000000E+01 0.
  .5000000000E+00  .1000000000E+01 0.                0.
 ROW 3
 0.               -.1000000000E+01  .2000000000E+01 -.1000000000E+01
 0.               0.                .1000000000E+01 0.
 ROW 4
 0,               0.               -.1000000000E+01  .2000000000E+01
 0.               0.               0.                 .1000000000E+01
 SPECIFY OPERATION (1,2).
#2
 I=
#2
 J=
#3
```

```
 M=
#-C[3,2]/C[2,2]
 ROW OPERATION COMPLETE
 THE CURRENT C-MATRIX IS
 ROW 1
  .2000000000E+01 -.1000000000E+01 0.               0.
  .1000000000E+01 0.               0.               0.
 ROW 2
 0.                .1500000000E+01 -.1000000000E+01 0.
  .5000000000E+00  .1000000000E+01 0.               0.
 ROW 3
 0.               0.                .1333333333E+01 -.1000000000E+01
  .3333333333E+00  .6666666667E+00  .1000000000E+01 0.
 ROW 4
 0.               0.               -.1000000000E+01  .2000000000E+01
 0.               0.               0.                .1000000000E+01
 SPECIFY OPERATION (1,2).
#2
 I=
#3
 J=
#4
 M=
#-C[4,3]/C[3,3]
 ROW OPERATION COMPLETE
 THE CURRENT C-MATRIX IS
 ROW 1
  .2000000000E+01 -.1000000000E+01 0.               0.
  .1000000000E+01 0.               0.               0.
 ROW 2
 0.                .1500000000E+01 -.1000000000E+01 0.
  .5000000000E+00  .1000000000E+01 0.               0.
 ROW 3
 0.               0.                .1333333333E+01 -.1000000000E+01
  .3333333333E+00  .6666666667E+00  .1000000000E+01 0.
 ROW 4
 0.               0.               0.                .1250000000E+01
  .2500000000E+00  .5000000000E+00  .7500000000E+00  .1000000000E+01
 WRITE A NUMBER OR EXPRESSION FOR DET(A).
 DET(A)=
#C[1,1]*C[2,2]*C[3,3]*C[4,4]
 DO YOU WANT TO SOLVE FOR THE INVERSE MATRIX B?
#YES
```

```
 USE BACK-SUBSTITUTION TO SOLVE FOR THE INVERSE B IN THE SYSTEM
 C[1,1] C[1,2] C[1,3] C[1,4]    B[1,1] B[1,2] B[1,3] B[1,4]
    0   C[2,2] C[2,3] C[2,4] X B[2,1] B[2,2] B[2,3] B[2,4]
    0      0   C[3,3] C[3,4]    B[3,1] B[3,2] B[3,3] B[3,4]
    0      0      0   C[4,4]    B[4,1] B[4,2] B[4,3] B[4,4]

    C[1,5] C[1,6] C[1,7] C[1,8]
 =  C[2,5] C[2,6] C[2,7] C[2,8].
    C[3,5] C[3,6] C[3,7] C[3,8]
    C[4,5] C[4,6] C[4,7] C[4,8]
 PRESS (RETURN) WHEN READY TO GIVE EXPRESSIONS FOR THE B[I,J].
#
 B[4,1]=

#C[4,4]/C[4,5]
      NO. TRY AGAIN.
 B[4,1]=

#C[4,5]/C[4,4]
 OK. B[4,1]=    .200000000000E+00
 B[3,1]=

#(C[3,5]-C[3,4]*B[4,1])/C[3,3]
 OK. B[3,1]=    .400000000000E+00
 B[2,1]=

#(C[2,5]-C[2,4]*B[4,1]:#-C[2,3]*B[3,1])/C[2,2]
 OK. B[2,1]=    .600000000000E+00
 B[1,1]=

#(C[1,5]-C[1,4]=#*B[4,1]-C[1,3]*B[3,1]-C[1,2]*B[2,1])/!
#C[1,1]
 OK. B[1,1]=    .800000000000E+00
 B[4,2]=

#C[4,6]/C[4,4]
 OK. B[4,2]=    .400000000000E+00
 B[3,2]=

#(C[3,6]-C[3,4]*B[4,2])/C[3,3]
 OK. B[3,2]=    .800000000000E+00
 B[2,2]=

#(C[2,5#6]-C[2,4]*B[4,2]-C[2,3]*B[3,2])/C[2,2]
 OK. B[2,2]=    .120000000000E+01
 B[1,2]=

#
#(C[1,6]-C[1,4]*B[4,2]-C[1,3]*B[3,2]=#-C[1,2]*B[2,2])/C[1,1]
 OK. B[1,2]=    .600000000000E+00
 B[4,3]=

#C[4,7]/C[4,4]
```

```
 OK. B[4,3]=    .600000000000E+00
 B[3,3]=

#(C[3,7]-C[3,4]*B[4,3])/C[3,3]
 OK. B[3,3]=    .120000000000E+01
 B[2,3]=

#
#(C[2,7]-C[2,4]*B[4,3]-C[2,3]*B[2,##3,3]-C[2,2]*B[2,3])/1
#
#(C[2,7]-C[2,4]*B[4,3]-C[2,3]*B[3,3])/C[2,2]
     NO. TRY AGAIN.
 B[2,3]=

#(C[2,7]-C[2,4]*B[4,3]-C[2,3]*B[3,3])/C[2,2]
 OK. B[2,3]=    .800000000000E+00
 B[1,3]=

#(C[1,7]-C[1,4]*B[4,3]-C[1,3]*B[3,3]-C[1,2]*B[2,3])/C[1,1]
 OK. B[1,3]=    .400000000000E+00
 B[4,4]=

#C[4,8]/C[4,4]
 OK. B[4,4]=    .800000000000E+00
 B[3,4]=

#(C[3,8]-C[3,4]*B[4,4])/C[3,3<#]
 OK. B[3,4]=    .600000000000E+00
 B[2,4]=

#(C[2,8]-C[2,4]*B[4,4]-C[2,3]*B[3,4])/C[2,2)#]
 OK. B[2,4]=    .400000000000E+00
 B[1,4]=

#(C[1,8]-C[1,4]*B[4,4]-C[2,3]###1,3]*B[3,4]-C[1,2]*B[2,4])/1
#C[1,1]
 OK. B[1,4]=    .200000000000E+00
 MULTIPLYING B*A, WE SHOULD GET THE IDENTITY MATRIX. ACTUALLY, B*A=
   .1000000000E+01 0.                 0.               0.
  0.                .1000000000E+01 0.               0.
  0.               0.                 .1000000000E+01 0.
  0.               0.                0.                 .1000000000E+01
 PROBLEM 2.
 SELECT THE SIZE OF MATRIX FOR THIS PROBLEM (2,3,4,NONE).
 N=
#NONE

 YOU HAVE TWO CHOICES

 (1)  TYPE "$LOGOFF" WHICH WILL SIGN YOU OFF THE COMPUTER.
 (2)  TYPE "$LESSON,-----" WHERE ----- IS THE SECTION NAME
      OF THE MATERIAL YOU WISH TO STUDY.

 CONSULT THE INDEX OF THE STUDENT MANUAL FOR THE LIST OF
 AVAILABLE SECTION NAMES.

#$LOGOFF
```

```
LESSON 19.  ROMBERG INTEGRATION        PRESS (RETURN) TO BEGIN.
#
 1.  INTRODUCTION

RECALL THE TRAPEZOIDAL RULE FOR APPROXIMATING THE DEFINITE
INTEGRAL I(F(X);[A,B])=I(F(X)) AS

******************************************************************

     T=(H/2)*(F[0]+2*(F[1]+...+F[N-1])+F[N])
WITH THE ERROR GIVEN AS
     E(H)=-(B-A)*(H↑2)*F"(Z)/12  WHERE  A=X[0]<X[N]=B.

******************************************************************

PRESS (RETURN) WHEN READY.
#
THE PURPOSE OF THIS LESSON WILL BE TO SHOW THAT FOR N SUBDIVISIONS
OF THE INTERVAL [A,B], WE MAY WRITE, FOR THE APPROXIMATION T, THE
EXPRESSION
     (1) T0[0]=I(F(X))+A*(H↑2)+O(H↑4) WHERE A IS A CONSTANT.
ONCE WE ESTABLISH (1), WE CAN USE 2*N SUBDIVISIONS, I.E. REPLACE
H BY H/2 TO OBTAIN YET ANOTHER APPROXIMATION,
     (2) T0[1]=I(F(X))+A*(H↑2)/4+O(H↑4).
WE THEN EXTRAPOLATE TO OBTAIN THE O(H↑4)-APPROXIMATION
     (3) T1[1]=(M*T0[1]-T0[0])/(M-1)=I(F(X))+O(H↑4).
WHAT VALUE OF M IS NEEDED.
M=
#4
OK
ALTHOUGH THE DEVELOPMENT OF (1) IS SOMEWHAT COMPLICATED, THE
STUDENT WILL DO WELL TO THOROUGHLY UNDERSTAND THE DEVELOPMENT IN
ORDER TO GAIN A DEEPER INSIGHT INTO THE CONCEPT OF EXTRAPOLATION.
PRESS (RETURN) TO CONTINUE.
#
2.  BASIC DIFFERENTIATION FORMULAS

IN DEVELOPING EQUATION (1) OF THE PREVIOUS SECTION, WE WILL USE
THE NUMERICAL DIFFERENTIATION FORMULAS FOR F'(X[0]), F"(X[0]), AND
F"'(X[0]) GIVEN BY
     (1) F'[0]=(F[1]-F[0])/H-(H/2)*F"[0]-(H↑2)*F"'[0]/6+O(H↑3)
     (2) F"[0]=(F[2]-2*F[1]+F[0])/(H↑2)-H*F"'[0]+O(H↑2)
     (3) F"'[0]=(F[3]-3*F[2]+3*F[1]-F[0])/(H↑3)+O(H)
WE WILL DERIVE THE FORMULAS BY SIMPLE TAYLOR SERIES.
PRESS (RETURN) WHEN READY.
#
EXPANDING ABOUT X[0],
     F(X)=F[0]+F'[0]*(X-X[0])+F"[0]*((X-X[0])↑2)/2
                +F"'[0]*((X-X[0])↑3)/6+F""(Z)*((X-X[0])↑4)/24
WHERE X[0]<Z<X[1].  EVALUATING AT X=X[1] GIVES
  (A) F[1]=F[0]+F'[0]*H+F"[0]*(H↑2)/2+F"'[0]*(H↑3)/6+O(H↑4)
WHICH DIFFERENTIATION FORMULA IS DERIVED DIRECTLY FROM (A)
(1,2,3,NONE)?
#1
```

```
 OK
 EVALUATING THE TAYLOR EXPANSION AT X=X[2] YIELDS
  (B) F[2]=F[0]+F'[0]*(2*H)+F"[0]*(4*H↑2)/2+F"'[0]*(8*H↑3)/6+O(H↑4)
           =F[0]+F'[0]*(2*H)+F"[0]*(2*H↑2)+F"'[0]*(4*H↑3)/3+O(H↑4)
 WHICH FORMULA IS DERIVED DIRECTLY FROM (B).
 (1,2,3,NONE)?
#2
     NO.  TRY AGAIN.  DON'T GUESS.
 (1,2,3,NONE)?
#NONE
 OK.  WE NEED MORE INFORMATION.
 SUPPOSE WE EVALUATE THE TAYLOR EXPANSION AT X=X[0]. WE OBTAIN
 THE IDENTITY
   (C) F[0]=F[0]
 WE CAN OBTAIN DIFFERENTIATION FORMULA 2 BY FORMING
     (M1*EQUATION (B) + M2*EQUATION(A) + M3*EQUATION(C))/(H↑2)
 DEFINE THE NUMERIC VALUES FOR M1, M2, AND M3.
 M1=
#1
 OK
 M2=
#-2
 OK
 M3=
#1
 OK
 NEXT, WE EVALUATE THE TAYLOR FORMULA AT X[3] TO OBTAIN
      (D) F[3]=F[0]+F'[0]*(3*H)+F"[0]*(9*H↑2)/2
                +F"'[0]*(27*H↑3)/6+O(H↑4).
 WE CAN OBTAIN DIFFERENTIATION FORMULA 3 BY FORMING
      (M1*EQ(A)+M2*EQ(B)+M3*EQ(C)+M4*EQ(D))/H↑3.
 DEFINE THE NUMERIC VALUES FOR M1,M2,M3, AND M4.
 M1=
#3
 OK
 M2=
#-3
 OK
 M3=
#-1
 OK
 M4=
#1
 OK
 WE NOW HAVE -3*EQ(A)+3*EQ(B)-EQ(C)+EQ(D).  DIVIDING BY H↑3
 GIVES DIFFERENTION FORMULA 3.  THE THREE DIFFERENTIATION FORMULAS
 WILL BE USED IN APPROXIMATING THE INTEGRAL.  PRESS (RETURN).
#
```

```
3.  GENERAL FORMULATION OF THE TRAPEZOIDAL RULE

 RECALL THAT OUR OBJECTIVE IS TO ESTABLISH THE GENERAL TRAPEZOIDAL
 RULE

 *****************************************************************

       TO[0]=I(F(X))+A*H↑2+O(H↑4)
 WHERE A IS CONSTANT AND TO[0] IS COMPUTED BY
      (H/2)*(F[0]+2*(F[1]+F[2]+...+F[N-1])+F[N]).

 *****************************************************************

 WE BEGIN BY INTEGRATING THE TAYLOR FORMULA
      F(X)=F[0]+(X-X[0])*F'[0]+((X-X[0])↑2)*F"[0]/2
           +((X-X[0])↑3)*F"'[0]/6+((X-X[0])↑4)*F""(Z)/24.
 THIS GIVES
      I(F(X);[X[0],X[1]])=A*F[0]+B*F'[0]+C*F"[0]+D*F"'[0]+E*F""(W)
 DEFINE A, B, C, D, AND E IN TERMS OF H.
 A=
#H
 OK
 B=
#H↑2/2
 OK.  SO B=(H↑2)/2.
 C=
#(H↑3)/6
 OK
 D=
#(H↑4)/24
 OK
 FORMING THE INTEGRAL OF THE LAST TERM, WE NOTE THAT ((X-X[0])↑4)
 DOES NOT CHANGE SIGN ON THE INTERVAL [X[0],X[1]].  SO WE MAY
 WRITE I((X-X[0])↑4*F""(Z)/24;[X[0],X[1]])
           =(F""(W)/24)*I((X-X[0])↑4);[X[0],X[1]]).
 E=
#(H↑5)/120
 OK
 SO, I(F(X);[X[0],X[1]])=
      =H*F[0]+(H↑2)*F'[0]/2+(H↑3)*F"[0]/6+(H↑4)*F"'[0]/24+O(H↑5).
 WE NOW REPLACE F'[0] BY THE 1ST DIFFERENTIATION FORMULA
      F'[0]=(F[1]-F[0])/H-(H/2)*F"[0]-(H↑2)*F"'[0]/6+O(H↑3)
 TO OBTAIN I(F(X);[X[0],X[1]])
      =H*F[0]+((H↑2)/2)*((F[1]-F[0])/H-(H/2)*F"[0]-(H↑2)*F"'[0]/6)
           +(H↑3)*F"[0]/6+(H↑4)*F"'[0]/24+O(H↑5)
      =(H/2)*(F[0]+F[1])-(H↑3)*F"[0]/12-(H↑4)*F"'[0]/24+O(H↑5)
 PRESS (RETURN).
#
```

```
 IN THE LAST FORMULA, REPLACE F"[0] BY THE 2ND DIFFERENTIATION
 FORMULA  F"[0]=(F[2]-2*F[1]+F[0])/(H↑2)-H*F"'[0]+0(H↑2) TO OBTAIN
 I(F(X);[X[0],X[1]])=
    (A) (H/2)*(F[0]+F[1])-(H/24)*(F[2]-2*F[1]+F[0])
                          +(H↑4)*F"'[0]/12+0(H↑5)
    (B) (H/2)*(F[0]+F[1])-(H/12)*(F[2]-2*F[1]+F[0])
                          +(H↑4)*F"'[0]/24+0(H↑5)
    (C) (H/2)*(F[0]+F[1])-(H/24)*(F[2]-2*F[1]+F[0])
                          +(H↑4)*F"'[0]/24+0(H↑5)
 CORRECT ANSWER IS (A,B,C,NONE)?
#B
 OK
 REPLACE F"'[0] IN (B) BY DIFFERENTIATION FORMULA (3)
      F"'[0]=(F[3]-3*F[2]+3*F[1]-F[0])/(H↑3)+0(H) TO OBTAIN
 I(F(X);[X[0],X[1]])=
      (A) (H/2)*(F[0]+F[1])-(H/12)*(F[2]-2*F[1]+F[0])
                             -(H/12)*(F[3]-3*F[2]+3*F[1]-F[0])+0(H↑5)
      (B) (H/2)*(F[0]+F[1])-(H/12)*(F[2]-2*F[1]+F[0])
                             -(H/24)*(F[3]-3*F[2]+3*F[1]-F[0])+0(H↑5)
      (C) (H/2)*(F[0]+F[1])-(H/12)*(F[2]-2*F[1]+F[0])
                             +(H/24)*(F[3]-3*F[2]+3*F[1]-F[0])+0(H↑5)
 CORRECT ANSWER IS (A,B,C,NONE)?
#C
 OK
 FOR THE GENERAL INTERVAL, WE HAVE
 I(F(X);[X[I],X[I+1]])
   =(H/2)*(F[I]+F[I+1])-(H/12)*(F[I+2]-2*F[I+1]+F[I])
                  +(H/24)*(F[I+3]-3*F[I+2]+3*F[I+1]-F[I])+0(H↑5)
 AND I(F(X);[A,B])=I(F(X);[X[0],X[N]])=
 (H/2)*SUM(F[I]+F[I+1])-(H/12)*SUM(F[I+2]-2*F[I+1]+F[I])
 +(H/24)*SUM(F[I+3]-3*F[I+2]+3*F[I+1]-F[I])+SUM(0(H↑5))
 WHERE SUM GOES OVER I=0,...,N-1.   PRESS (RETURN) WHEN READY.
#
 WE RECOGNIZE SUM(F[I]+F[I+1]) AS THE TRAPEZOIDAL APPROXIMATION
     F[1]+2*(F[2]+...+F[N-1])+F[N].
 ALSO, THE LAST SUM,
 SUM(0(H↑5))=N*(0(H↑5))=((B-A)/H)*0(H↑5)=0(WHAT?)
#H↑4
 OK
 IN TERMS OF THE F[I] FOR THE APPROPRIATE VALUES OF I, WRITE AS A
 FOUR TERM EXPRESSION
 SUM(F[I+2]-2*F[I+1]+F[I])=
#F[N+1]-F[N]-F[1]+F[0]
 OK.  SUM(F[I+2]-2*F[I+1]+F[I])=F[0]-F[1]+F[N+1]-F[N].
 USING SIX TERMS,
 SUM(F[I+3]-3*F[I+2]+3*F[I+1]-F[I])=
#F[N+2]+#-2*F[N+1]+F[N]-F[2]+2*F[2#2#1]-F[0]
 OK
 WE NOW HAVE I(F(X);[A,B])=
 (F[0]+2*(F[1]+...+F[N-1])+F[N])-(H/12)*(F[0]-F[1]+F[N+1]-F[N])
            +(H/24)*(F[N+2]-2*F[N+1]+F[N]-F[2]+2*F[1]-F[0])+O(H↑4).
 USING DIFFERENTIATION FORMULAS 1 AND 2 OF SECTION 2,
      F[0]-F[1]=-H*F'[0]-(H↑2)*F"[0]/2+O(H↑3)
      F[N+1]-F[N]=H*F'[N]+(H↑2)*F"[N]/2+O(H↑3)
      F[2]-2*F[1]+F[0]=(H↑2)*F"[0]+(H↑3)*F"'[0]+O(H↑4)
      F[N+2]-2*F[N+1]+F[N]=(H↑2)*F"[N]+(H↑3)*F"'[N]+O(H↑4)
 MAKING THESE SUBSTITUTIONS AND COLLECTING TERMS,
 I(F(X);[A,B])=
  (A) (H/2)*SUM(F[I]+F[I+1])+O(H↑4)
  (B) (H/2)*SUM(F[I]+F[I+1])-(H↑2)*(F'[N]-F'[0])/12+O(H↑4)
  (C) (H/2)*SUM(F[I]+F[I+1])-(H↑2)*(F'[N]-F'[0])/12
                            +(H↑3)*(F"[N]-F"[0])/12 + O(H↑4)
 CORRECT ANSWER IS (A,B,C)?
#
#A
      SINCE THE TRAPEZOIDAL RULE IS AN O(H↑2)-APPROXIMATION TO THE
      INTEGRAL, WE KNOW (A) IS WRONG.  SEE LESSON 18.
 CORRECT ANSWER IS (A,B,C)?
#B
 OK
 THIS ESTABLISHES THE BASIC FORMULA FOR THE TRAPEZOIDAL RULE.

 *****************************************************************

      TO[0]=I(F(X);[A,B])-(H↑2)*(F'[N]-F'[0])+O(H↑4)

 *****************************************************************

 WHERE THE COMPUTATION IS TO[0]=F[0]+2*(F[1]+...+F[N-1])+F[N].
 SINCE X[0]=A AND X[N]=B REGARDLESS OF THE NUMBER OF SUBDIVISIONS
 N, WE HAVE F'[N]-F'[0]=C IS CONSTANT.
 PRESS (RETURN) WHEN READY.
#
 4.  ROMBERG INTEGRATION -- SIMPLE EXTRAPOLATION

 WE HAVE FOR N SUBDIVISIONS OF THE INTERVAL [A,B],
      (A) TO[0]=I+C*(H↑2)+O(H↑4)
 AND FOR 2*N SUBDIVISIONS, (I.E. REPLACE H BY H/2),
      (B) TO[1]=I+C*(H↑2)/4+O(H↑4)
 IN TERMS OF TO[0] AND TO[1], WRITE AN O(H↑4) APPROXIMATION TO I
 T1[1]=I+O(H↑4)=
#4*TO[0]-TO[1]
      TRY AGAIN OR TYPE HELP.
 T1[1]=I+O(H↑4)=
#(4*TO[1]-T[#O[0])/3
 OK
 EXAMPLE 4A.  SUPPOSE WE WISH TO APPROXIMATE I(1/X;[1,2]).
 FOR H=.5, WE HAVE  (X[0],F[0])=(1,1), (X[1],F[1])=(1.5,2/3), AND
 (X[2],F[2])=(2.0,.5).  WRITE THE O(H↑2)-TRAPEZOIDAL APPROXIMATION
 TO[0]=
#.25*(1+4/3+.5)
```

```
 OK
 TO[0]=    .708333333333329E+00
 IN ORDER TO FORM T1[0], WE USE 4 SUBDIVISIONS WITH H REPLACED BY
 (H/2)=.25. THIS YIELDS THE NUMERIC VALUES
 X[0]=
#1
 OK
 X[3]=
#2/3
      TRY AGAIN.
 X[3]=
#1.5
      X[3]=1.75
 F[1]=
#.8
 OK
 WRITE THE O(H↑2)-TRAPEZOIDAL RULE
 TO[1]=
#.125*(1+2*(1/1.25+1/1.5+1/1.75)+.5)
 OK
 TO[1]=    .697023809523799E+00
 USE SIMPLE EXTRAPOLATION AND WRITE THE O(H↑4)-APPROXIMATION.
 T1[1]=
#(4*.697023809523799-.708333333333329)/3
 OK
 T1[1]=    .693253968253956E+00

 5.  ROMBERG INTEGRATION - REPEATED EXTRAPOLATION

 WITH A LITTLE TIME AND EFFORT, WE COULD HAVE CARRIED MORE TERMS IN
 OUR TAYLOR EXPANSION TO OBTAIN
      (A) TO[0]=I+C1*(H↑2)+C2*(H↑4)+O(H↑6) WITH C1 AND C2 CONSTANT.
 REPLACING H BY H/2, WE OBTAIN TO[1]=I+D1*(H↑2)+D2*(H↑4)+O(H↑6)
 IN TERMS OF C1 AND C2,
 D1=
#C1/4
 D2=
#C2/16
 OK
      (B) TO[1]=I+C1*(H↑2)/4+C2*(H↑4)/16+O(H↑6)
 REPLACING H BY H/4 IN (A), TO[2]=I+E1*H↑2+E2*H↑4+O(H↑6).
 IN TERMS OF C1 AND C2,
 E1=
#C1/16
 OK
 E2=
#C2/54##64##256
 OK
      (C) TO[2]=I+C1*(H↑2)/16+C2*(H↑4)/256+O(H↑6)
 WE NOW USE SIMPLE EXTRAPOLATION ON (A) AND (B) TO OBTAIN THE
 O(H↑4)-APPROXIMATION T1[1]=(4*TO[1]-TO[0])/3=I+F1*(H↑4)+O(H↑6)
 F1=
#-C2/4
#-3*C2/4
```

```
      TRY AGAIN.
 F1=
#-C2/4
 OK
 USE SIMPLE EXTRAPOLATION (B) AND (C) TO OBTAIN ANOTHER
 O(H↑4)-APPROXIMATION T1[2]=(4*T0[2]-T0[1])/3=I+F2*(H↑4)+O(H↑6).
 F2=
#-C2/64
 OK
 WE NOW HAVE THE TWO O(H↑4)-APPROXIMATIONS
      (D) T1[1]=I-(H↑4)*C2/4+O(H↑6)
      (E) T1[2]=I-(H↑4)*C2/64+O(H↑6)
 BY SIMPLE EXTRAPOLATION ON (D) AND (E), WE HAVE FOR THE PROPER M,
      (F) T2[2]=(M*T1[2]-T1[1])/(M-1)=I+O(H↑6)
 M=
#16
 OK
 EXAMPLE 5A.  IN EXAMPLE 4A, WE WANTED TO APPROXIMATE
 I(1/X;[1,2]) AND WE CALCULATED FOR FOR H=.5, N=2
 T0[0]=    .708333333333329E+00
 T0[1]=    .697023809523802E+00  T1[1]=    .693253968253959E+00
 FOR 4*N, H/4, CALCULATE
 T0[2]=
#.0625*(1+2*(1/1.125+1/1.25+1/1.375+1/1.5+1/1.625+1/1.75+!
#1/1.875)+.5)
 OK
 T0[2]=    .694121850371840E+00
 PERFORM THE EXTRAPOLATIONS
 NOTE THAT T1[1] WAS COMPUTED IN EXAMPLE 4A.
 T1[2]=
#(4*T0[2]-T0[1])/3
 OK
 T1[2]=    .693154530654514E+00
 T2[2]=
#(16*T1[2]-T1[1])/15
 OK
 T2[2]=    .693147901481215E+00

 6.  ROMBERG INTEGRATION -- EXTRAPOLATION TO THE LIMIT

 FORM THE FOLLOWING TABLE OF VALUES
      T0[0]
      T0[1]  T1[1]
      T0[2]  T1[2]  T2[2]
        .      .      .      .
      T0[K]  T1[K]  T2[K]  . . . TK[K]
 WHERE T0[I] IS THE TRAPEZOIDAL APPROXIMATION WITH (I+1)*N
 SUBDIVISIONS AND SPACING H/(2↑I).  T1[I] ARE 1ST EXTRAPOLATIONS
 WITH MULTIPLIER M=4, T2[I] ARE 2ND EXTRAPOLATIONS WITH M=4↑2, AND
 TJ[I] ARE JTH EXTRAPOLATIONS USING M=4↑J.   PRESS (RETURN).
#
 THIS IS THE END OF LESSON 19.
 DO YOU WISH TO BEGIN THE PROBLEM MODE?
#YES
```

```
 PROBLEM MODE FOR LESSON 19

 FOR EACH PROBLEM, YOU MUST DEFINE THE TRAPEZOIDAL FORMULAS FOR THE
 SPECIFIED VALUES OF N AND THE FORMULAS FOR EXTRAPOLATION TO THE
 LIMIT.  YOU MAY TERMINATE A PROBLEM ANYTIME BY TYPING 'STOP'.

 YOUR PROBLEM SELECTION IS (1,2,NONE)?
#2
 PROBLEM 2.  WE WISH TO USE ROMBERG INTEGRATION TO APPROXIMATE
 I(LN(X);[1,3]) USING N=1,2,4, AND 8.

  I  X[I]           F[I]

  0 1.00  0.
  1 1.25   .223143551314203E+00
  2 1.50   .405465108108153E+00
  3 1.75   .559615787935410E+00
  4 2.00   .693147180559929E+00
  5 2.25   .810930216216310E+00
  6 2.50   .916290731874131E+00
  7 2.75   .101160091167845E+01
  8 3.00   .109861228866808E+01
 WRITE THE EXPRESSIONS FOR THE TRAPEZOIDAL RULES USING THE TABLE.
 FOR N=1, T0[0]=
#F[8]+F[0]
 OK. T0[0]=    .109861228866808E+01
 FOR N=2, T0[1]=
#.5*(2*F[4]+F[8])
 OK. T0[1]=    .124245332489397E+01
 FOR N=4, T0[2]=
#.25*(2*(F[2]+F[4]+F[6])+F[8])
 OK. T0[2]=    .128210458243812E+01
 FOR N=8, T0[3]=
#
#.125*(2*(F[2]+F[1]+F[3]+F[4]+F[5]+F[6]+F[7]);F[8])######+F[8])
 OK. T0[3]=    .129237490800514E+01
 DEFINE THE O(H↑4)-APPROXIMATIONS
 T1[1]=
#(4*T0[1]-T0[0])/3
 OK. T1[1]=    .129040033696926E+01
 T1[2]=
#(4*T0[2]-T0[1])/3
 OK. T1[2]=    .129532166828617E+01
 T1[3]=
#(4*T0[3]-T0[2])/3
 OK. T1[3]=    .129579834986081E+01
 DEFINE THE O(H↑6)-APPROXIMATIONS.
 T2[2]=
#(16*T1[2]-T1[1])/15
 OK. T2[2]=    .129564975704062E+01
 T2[3]=
#(16*T1[3]-T1[2])/15
```

```
 OK. T2[3]=    .129583012863245E+01
 DEFINE THE O(H↑8)-APPROXIMATION.
 T3[3]=
#(64*T2[3]-T2[2])/63
 OK. T3[3]=    .129583299167358E+01
 THE ROMBERG EXTRAPOLATION TABLE IS
 H=    .200000000000000E+01
   .109861228866808E+01
 H=    .100000000000000E+01
   .124245332489397E+01    .129040033696926E+01
 H=    .500000000000000E+00
   .128210458243812E+01    .129532166828617E+01    .129564975704062E+01
 H=    .250000000000000E+00
   .129237490800514E+01    .129579834986081E+01    .129583012863245E+01
   .129583299167358E+01
 SELECT ANOTHER PROBLEM.
 YOUR PROBLEM SELECTION IS (1,2,NONE)?
#1
 PROBLEM 1.  WE WISH TO USE ROMBERG INTEGRATION TO APPROXIMATE
 I(SIN(X)/X;[0,1]) USING N=1,2, AND 4 SUBDIVISIONS.  (NOTE, AT X=0,
 SIN(X)/X=1 SINCE THIS VALUE AGREES WITH THE LIMIT.)

  I  X[I]          F[I]

  0 0.00   .100000000000000E+01
  1  .25   .989615837018093E+00
  2  .50   .958851077208401E+00
  3  .75   .908851680031113E+00
  4 1.00   .841470984807891E+00
 WRITE THE EXPRESSIONS FOR THE TRAPEZOIDAL RULES USING THE TABLE.
 FOR N=1, T0[0]=
#.5*(F[0]+F[4])
 OK. T0[0]=    .920735492403946E+00
 FOR N=2, T0[1]=
#.25*(F[0]+2*F[2];F[4])######+F[4])
 OK. T0[1]=    .939793284806171E+00
 FOR N=4, T0[2]=
#.125*(F[0]+2*(F[1]+F[2]+F[3])+F[4])
 OK. T0[2]=    .944513521665385E+00
 DEFINE THE O(H↑4)-APPROXIMATIONS
 T1[1]=
#(4*T0[1]-T0[0])/3
 OK. T1[1]=    .946145882273576E+00
 T1[2]=
#(4*T0[2]-T0[1])/3
 OK. T1[2]=    .946086933951790E+00
 DEFINE THE O(H↑6)-APPROXIMATIONS.
 T2[2]=
#
#(16*T1[2]-T1[1])/15
```

```
 OK. T2[2]=   .946083004063670E+00
 THE ROMBERG EXTRAPOLATION TABLE IS
 H=   .100000000000000E+01
   .920735492403946E+00
 H=   .500000000000000E+00
   .939793284806171E+00   .946145882273576E+00
 H=   .250000000000000E+00
   .944513521665385E+00   .946086933951790E+00   .946083004063670E+00
 SELECT ANOTHER PROBLEM.
 YOUR PROBLEM SELECTION IS (1,2,NONE)?
#NONE

 YOU HAVE TWO CHOICES

 (1)  TYPE "$LOGOFF" WHICH WILL SIGN YOU OFF THE COMPUTER.
 (2)  TYPE "$LESSON,-----" WHERE ----- IS THE SECTION NAME
      OF THE MATERIAL YOU WISH TO STUDY.

 CONSULT THE INDEX OF THE STUDENT MANUAL FOR THE LIST OF
 AVAILABLE SECTION NAMES.

#$LOGOFF
```